Distributed Optical Raman Amplifiers in Ultra High Speed Long Haul Transmission Optical Fiber Telecommunication Networks

Abd El–Naser A. Mohammed[1], Mohamed A. Metawe'e[2]
Ahmed Nabih Zaki Rashed[3], and Mahmoud M. A. Eid[4]

[1,2,3,4]ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING DEPARTMENT
Faculty of Electronic Engineering, Menouf 32951, Menoufia University, EGYPT
[1]E-mail: Abd_elnaser6@yahoo.com, [3]E-mail: ahmed_733@yahoo.com

Abstract: ***In the present paper, we have investigated parametrically and numerically the new trends and progress of distributed optical Raman amplifiers in ultra high speed ultra long haul transmission optical telecommunication networks. Moreover, we have been modeled and analyzed the development of distributed fiber Raman amplifiers and its application in the transmission of light-wave modulated signals in optical wavelength division multiplexing (WDM), and dense wavelength division multiplexing (DWDM) systems. We have developed the numerical techniques based on mathematical laboratory (MATLAB) programming to solve the coupled wave equations to obtain design parameters, operational characteristics, and its implementation in the light-wave modulated envelop propagation equations for transmission of multi Gbit/sec signals in DWDM optical systems. Also in the same way, we have deeply studied the transmission distances and transmission bit rates and products either per channel or per link within Raman amplification technique in forward, backward, and bi-directional pumping direction configurations through standard single-mode fiber (SSMF) pure silica fiber cable core material using Soliton transmission technique for upgrading optical access network performance to provide maximum data transfer for the supported number of users.***

Keywords: Noise figure (NF), Distributed optical Raman amplifiers, Ultra long haul transmission, Soliton technique.

1. Introduction

A network planner needs to optimize the various electrical and optical parameters to ensure smooth operations of a wavelength division multiplexing (WDM) network [1]. To the networking world, the optical layer (WDM layer) appears as a barren physical layer whose function is to transport raw bits at a high bit rate with negligible loss [2]. Until the bit rate and the transmission distance is under some bounded constraint, it is often not important to consider the optical parameters. However, as the bit rate increases and transmission length increases, these optical parameters have the capability of playing truant in the network. A network planner must consider the affecting parameters and build a network that accommodates the impairments caused by the optical parameters [3]. The fiber Raman optical amplifier is quickly emerging as an important part of long-distance, high-capacity, and ultra high-speed optical communication systems [4]. In modern long haul fiber-optic communication systems, the transmission distance is limited by fiber loss and dispersion. Traditional methods to overcome this limitation, which use electrical conversion of the optical signal, such as repeaters to retransmit signals at progressive stages are becoming increasingly complex and expensive. The ability to use any type of fiber as a gain medium has utilized for Distributed Raman Amplification (DRA), where amplification takes place inside the transmission fiber itself. However, Raman technology can also be applied to lumped amplifiers, in which case the ability to amplify signals in any wavelength band can be used in wavelength bands, high capacity all-Raman optical links have been deployed combining distributed fiber Raman amplifiers (DRAS). As the optical signal moves along a standard single mode fiber SSMF, it gets attenuated along the fiber and if the data speed is high enough (> 10 Gbit/sec). The bandwidth of optical fibers is really great if the S-band, C-band, and L-band are utilized efficiently. The important feature of Raman amplifiers is its gain bandwidth, which is determined by pump wavelength. Multi-wavelength pumping scheme is usually used to increase the gain flattening and bandwidth for high capacity WDM transmission systems. In backward-pumped fiber Raman amplifiers, other noise sources, such as the relative intensity noise (RIN) transfer are minimized [5].

In the present study, we have been modeled parametrically the different pumping direction configurations to provide flexibility in the optical system for distributed fiber Raman Amplifiers for employment in ultra high speed long haul transmission optical access network applications. Moreover, we have deeply studied the distributed fiber Raman amplifiers with the transmission fibers, and pumped at any wavelength to provide wide gain bandwidth and improve signal to noise ratio of the transmitted optical signals to allow both long transmission distance and high capacity networks. We have developed the investigations of distributed fiber Raman amplifiers for different transmission silica fibers for integration and adaptation of the gain and noise profile for signal propagation over ultra long haul transmission bit rates and distance in DWDM optical telecommunication networks. Also, the investigation of the soliton bit rate and product either per link or per channel in all pumping direction configurations is deeply studied over wide range of the affecting set of the parameters.

2. Simplified DWDM Optical Network Architecture Model

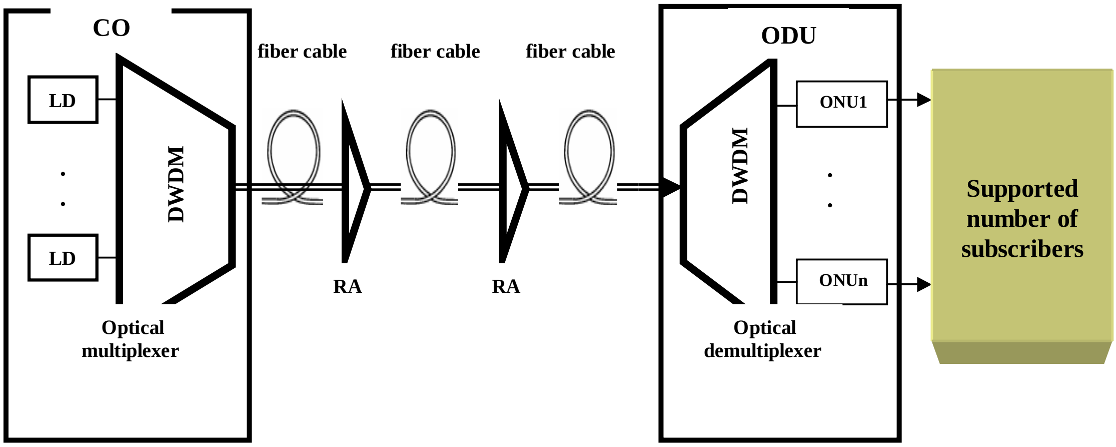


Figure1. DWDM optical telecommunication network architecture model.

The enormous growth in the demand of bandwidth is pushing the utilization of fiber infrastructures to their limits. To fulfill this requirement the constant technology evolution is substituting the actual signal wavelength systems connected in a point to point technology by dense wavelength division multiplexing (DWDM) systems, creating the foundations for the optical transport network (OTN). The objective is the deployment of a optical network layer with the same flexibility because it is more economical and allows a better performance in the bandwidth utilization. The optical demultiplexer which divides the light beam in to different optical channels adjustable at different specific wavelengths and then directed to optical network units (ONUs) and finally directed to the minimum or maximum number of supported users or subscribers depend on the process of add or drop multiplexing. A DWDM system can be described as a parallel set of optical channels, each using a slightly different wavelengths, but all sharing a single transmission medium or fiber. Fig. 1 illustrates the functionality of a multi-channel DWDM transmission system when various 10 Gbit/sec signals are fed to optical transmission modules. An optical DWDM coupler (multiplexer) then bunches these optical signals together on one fiber and forwards them as a multiplexed signal to an optical fiber Raman amplifier. Depending on path length and type of fiber used, one or more optical fiber Raman amplifiers can be used to boost the optical signal for long fiber links. At termination on the receiving end, the optical signals are preamplified, then separated using optical filters (demultiplexers) before being converted into electrical signals in the receiver modules. For bi-directional transmission, this produce must be duplicated in the opposite direction to carry the signals in that particular direction.

3. Schematic Basic Configuration of Optical Raman Amplifier

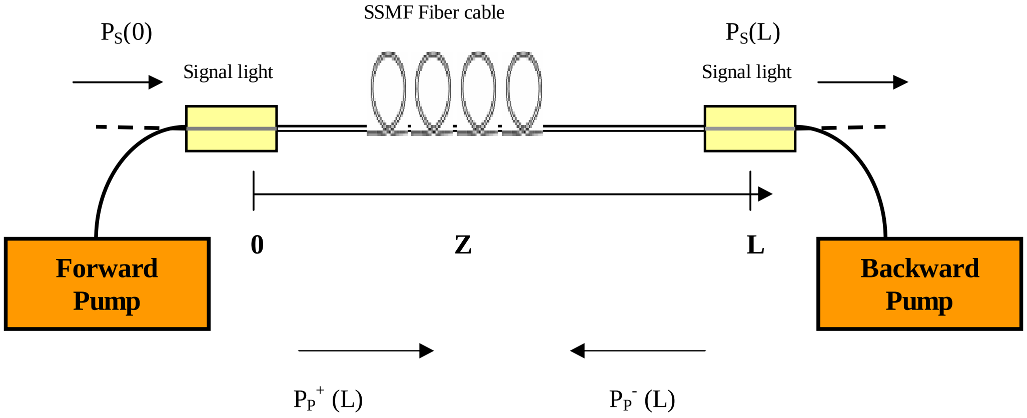


Figure 2. Schematic configuration of typical optical Raman amplifier.

Figure 2 is a schematic showing the configuration of a typical fiber Raman amplifier. Basically, pumping light and signal light are input to a single amplifier fiber and amplification is effected by means of the stimulated scattering that occurs in the fiber. Fig. 2, shows a configuration in which pumping light propagates bidirectionally in the Raman amplifier fiber, but in some it propagates in the same direction as the light signal (forward pumping) or the opposite direction (backward pumping). Generally, speaking with forward pumping the signal to noise ratio (SNR) can be kept high, while with backward pumping the saturation output power can be increased. In the case of a Raman amplifier the process of optical amplification takes place so rapidly that, unless the intensity noise of the forward pumping light is sufficiently small, the pumping light noise will be transferred to the signal light resulting in increasing transmission bit error rates. Thus in many cases only backward pumping is used. Raman scattering occurs in any silica glass which means if we inject an optical beam (pump) in an optical fiber, then a signal passing through that fiber will be amplified if its frequency is around the shifted frequency of the pump. This is called Stock shift, which is a round 13 GHz from the pump propagating beam frequency assuming that its wavelength is 1.45 μm [6]. In distributed types, amplification occurs all along the fiber between say two stations with the pump placed either near the transmitter in which case it is called forward pumping or near the receiver in which case it is called backward pumping. The forward pumping direction provides the lowest noise figure. In fact, the noise is sensitive to the gain and the gain is the highest when the input power is the lowest. Backward pumping provides the highest saturated output power. Bi-directional pumping scheme has a higher performance than the other two by combining the lowest noise figure and the highest output power advantageous although it requires two pump lasers. In addition, in this scheme the small signal gain is uniformly distributed along the whole active fiber [7].

4. The System Model and Equations Analysis

It is assumed that the power of the first pump source is S P_P and the second source pump is (1-S) P_P respectively, where P_P is the pump power and S is a coefficient showing the power that is being pumped in the signal direction. The evolution of the signal power (P_s) and the power of the pump source propagating along the single mode optical fiber can be quantitatively described by different equations called propagation equations. The signal and pump power can be expressed as [8]:

$$\pm\frac{dP_P}{dz} = -\frac{v_P}{v_S} g_R P_P P_S - \alpha_P P_P \quad , \qquad (1)$$

$$\frac{dP_S}{dz} = g_R P_P P_S - \alpha_S P_S \quad , \qquad (2)$$

Where g_R in $W^{-1}m^{-1}$is the Raman gain coefficient of the fiber cable length, α_S and α_P are the attenuation of the signal and pump power in silica-doped fiber, v_S and v_P are the signal and pump frequencies. The signs of "+" or "-" are corresponding to forward and backward pumping. Since P_P » P_S, the first term in Eq. (1) is negligibly low compared with the second and its influence can be neglected. Therefore, Eq. (1) can be solved when both sides of the equation are integrated. When using forward pumping (S=1), the pump power can be expressed as the following expression [8]:

$$P_P(z) = P_P(0)\exp(-\alpha_P L) \quad , \qquad (3)$$

In the backward pumping case (S=0) the pump power is respectively equal to:

$$P_P(z) = P_P(0)\exp[-\alpha_P(L-z)] \quad , \qquad (4)$$

When a bi-directional pumping [9] is used (S =0-1) the laser source work at the same wavelength at different pump power. To calculate the pump power at point z it can be used:

$$P_P(z) = SP_P(0)\exp(-\alpha_P L) + (1-S)P_P(0)\exp[-\alpha_P(L-z)], \qquad (5)$$

If the values of P_P are substituted in differential Eq. (2), and it is integrated from 0 to L for the signal power in the forward and the backward pumping can be written as:

$$P_S(L) = P_S(0)\exp\left[g_R\, S\, P_0\,\frac{1-\exp(-\alpha_P z)}{\alpha_P} - \alpha_S z\right] = G_F P_S(0), \; (6)$$

$$P_S(L) = P_S(0)\exp\begin{bmatrix} g_R(1-S)P_0\, x\,\dfrac{\exp(-\alpha_P L)(\exp(\alpha_P z)-1)}{\alpha_P} \\ -\alpha_S z \end{bmatrix} \qquad (7)$$
$$= G_B P_S(0),$$

Where G_F, G_B are the net gain in the forward and backward pumping respectively. The net gain is one of the most significant parameters of fiber Raman amplifiers. It describes the signal power increase in the end of the transmission span and presents the ratio between the amplifier accumulated gain and signal loss. It can be simply described by the following expression:

$$G_{net}(L) = {P_S(L)}/{P_S}(0) \quad , \qquad (8)$$

With P_0 being the pump power at the input end. Hence the signal intensity at output of amplifier, fiber cable length L is determined by the following expression [9]:

$$P_S(L) = P_S(0)\exp\left(\frac{g_R P_0 L_{eff}}{A_{eff}} - \alpha_S L\right) \quad , \qquad (9)$$

The effective length, L_{eff}, is the length over which the nonlinearities still holds or Stimulated Raman scattering (SRS) occurs in the fiber and is defined as:

$$L_{eff} = \frac{1-\exp(-\alpha_P L)}{\alpha_P} \quad , \qquad (10)$$

Hence the amplification gain defined as the ratio of the power signal with and without Raman amplification, is given by the following expression:

$$G_A = \frac{P_S(L)}{P_S(0)\exp(-\alpha_S L)} = \exp(g_0 L), \qquad (11)$$

This is referred to as the on-off Raman gain, where g_0 is the small signal gain used in amplification technique. The spontaneous scattering factor is dependent on the temperature of the amplifier and is defined as [10]:

$$n_{sp} = \frac{1}{1-\exp\left(\dfrac{-h\,\Omega_R}{k_B T}\right)} \quad , \qquad (12)$$

Where h is the Planck's constant, k_B is the Boltzman's constant, T is the temperature of the Raman amplifier, and Ω_R is the Raman gain coefficient at the Stokes frequency shift. For a fiber Raman amplifier n_{sp} ≈1.13 as a fully inversion of the stimulated carrier amplification process always happens. The noise figure (NF) is the determination of the signal denigration over the length of the transmission span. It is the signal to noise ratio of input over that of the output and in fiber Raman amplifier. It is dependent upon the pumping power and the gain of the optical system as:

$$Noise\,Figure(NF) = 2n_{sp}\frac{g_R}{A_{eff}}\int_0^L \frac{P_P\,dz}{G(z)}dz + \frac{1}{G_{net}(L)} \quad , \qquad (13)$$

Where G (z) is the net gain at distance z along the fiber cable length, A_{eff} is the effective area of the fiber cable core, and G_{net} (L) is the net gain at the end of the fiber cable length. The maximum transmit power per channel, as a function of fiber cable link length can be expressed as [11]:

$$P_{Transmitted} \langle \frac{4\times10^4}{N_{ch}(N_{ch}-1)\Delta\lambda_S\,L} \quad , \qquad (14)$$

Where N_{ch} be the number of channels, $\Delta\lambda_S$ be the channel spacing in nm, and L is to be the length of the fiber cable link in km. The maximum transmitted power per channel deceases. If the spacing is fixed, the power launched decreases with N_{ch} inversely with a square term [11]. The standard single mode fiber cable is made of the pure silica material which the investigation of the spectral variations of the waveguide refractive-index (n) require Sellemeier equation under the form [20]:

$$n^2 = 1 + \frac{B_1\lambda^2}{\lambda^2 - B_2^2} + \frac{B_3\lambda^2}{\lambda^2 - B_4^2} + \frac{B_5\lambda^2}{\lambda^2 - B_6^2} \qquad (15)$$

The parameters of Sellmeier equation coefficients for pure silica material, as a function of ambient temperature as [12]: B_1= 10.668422193, B_2= 0.0301516485 * $(T/T_0)^2$, B_3= 3.043474218 x 10^{-3}, B_4= 1.1347511235 * $(T/T_0)^2$, B_5= 1.54133408, B_6= 1.104 x 10^3. Where T is the temperature of the material, °C, and T_0 is the reference temperature and is considered as 27 °C. Then the second differentiation of Eq. (15) w. r. t λ yields:

$$\frac{d^2n}{d\lambda^2} = \frac{1}{n}\left[\begin{array}{l}\frac{B_1(\lambda^2 - B_2^2) - 4\lambda^2}{(\lambda^2 - B_2^2)^3} + \frac{B_3(\lambda^2 - B_4^2) - 4\lambda^2}{(\lambda^2 - B_4^2)^3} + \\ \frac{B_5(\lambda^2 - B_6^2) - 4\lambda^2}{(\lambda^2 - B_6^2)^3}\end{array}\right] , \qquad (16)$$

The total chromatic dispersion coefficient D_t is given by:

$D_t = D_m + D_w$ (17)

Both D_m, and D_w are given by (for the fundamental mode):

$$D_m = -\frac{\lambda}{c}\left(\frac{d^2n}{d\lambda^2}\right), \quad n\sec/nm.km \qquad (18)$$

$$D_w = -\left(\frac{n_{cladding}}{c\,n}\right)\left(\frac{\Delta n}{\lambda}\right)Y \quad , \quad n\sec/nm.km \qquad (19)$$

Where c is the velocity of the light, 3 x 10^8 m/sec, n is the refractive-index of the fiber cable core, Y is a function of wavelength, where the relative refractive-index difference is:

$$\Delta n = \frac{n - n_{cladding}}{n} \quad , \qquad (20)$$

In any infinitesimal segment of fiber, dispersion on one hand and non linearity of the refractive-index on the other hand produce infinitesimal modulation angles which exactly compensate reciprocally. Under such conditions the pulse shape is the same everywhere. All this provided that a soliton waveform be used with a peak power [13]:

$$P_1 = \frac{\Delta\lambda^3 D_t A_{eff}}{4\pi^2 c\, n_2\, t_0^2} \; , \qquad (21)$$

Where n_2 is the nonlinear Kerr coefficient, 2.6 x 10^{-20} m^2/Watt, Δλ is the spectral line width of the optical source in nm, P_1 is the peak power in watt, A_{eff} is the effective area of the cable core fiber in μm^2, D_t is the total chromatic dispersion coefficient in nsec/nm.km. Then the pulse intensity width in nsec is given by:

$$t_0 = \sqrt{\frac{\Delta\lambda^3 D_t\, A_{eff}}{4\pi^2\, P_1\, n_2\, c}} \quad , \quad n\sec \qquad (22)$$

Then the Soliton transmission bit rate per optical network channel is given as follows [14]:

$$B_{rsc} = \frac{1}{10t_0} = \frac{0.1}{t_0}, \quad Gbit/\sec/channel \qquad (23)$$

Then the Soliton transmission bit rate per link is given:

$$B_{rsl} = \frac{0.1 * N_{link}}{t_0}, \quad Gbit/\sec/link \qquad (24)$$

Also in the system model analysis, the transmitted channels per link is given by the following expression:

$$\Delta N_{ch} = \frac{N_{ch}}{N_L} \quad , \qquad (25)$$

Where N_{Link} is the total number of links in the fiber cable core, and N_{ch} is the total number of channels. The available soliton transmitted bit rate B_{rs} is compared as the fiber cable length, L, and consequently the soliton product P_{rsc} per channel is computed as:

$$P_{rsc} = B_{rsc} * L, \quad Tbit.km/\sec \qquad (26)$$

Also, in the same way, the soliton product P_{rsl} per link is computed as the following expression:

$$P_{rsl} = B_{rsl} * L, \quad Tbit.km/\sec \qquad (27)$$

5. Results and Discussions

In the analysis of our results, we have investigated parametrically and numerically the new trends of distributed fiber Raman amplifiers in ultra high speed ultra long haul transmission optical telecommunication networks in the interval of 1.45 μm to 1.65 μm under the set of affecting parameters of room temperature (27 °C). The following set of the data of system model are employed to obtain the best performance characteristics of distributed optical fiber Raman amplifiers in optical telecommunication networks for different pumping direction configurations as follows:

$1.5 \le \lambda_{si}$, optical signal wavelength, $\mu m \le 1.65$, $1.4 \le \lambda_p$, pumping wavelength, $\mu m \le 1.55$, α_{si}=0.2 dB/km, α_P=0.35 dB/km, Pumping power: P_P= 0.25 Watt/pump, $2 \le P_{si}$, optical signal power, mwatt ≤ 20, T=T_0= 27 °C, A_{eff}= 85 μm^2, N_L: total number of links up to 24 links, $\Delta\lambda_s$ =0.2 nm, $0.001 \le \Delta n$, relative refractive-index difference ≤ 0.009, N_t: total number of channels up to 600 channels, Raman gain coefficient: g_R=0.7 W^{-1}. km^{-1}.

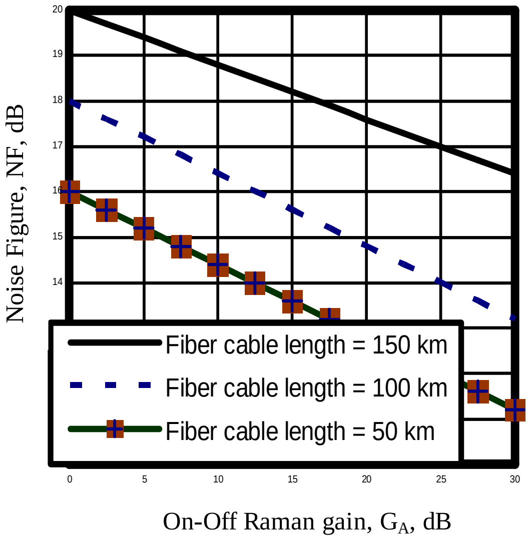


Figure 3. Variations of the noise figure with the on-off Raman gain at the assumed set of parameters.

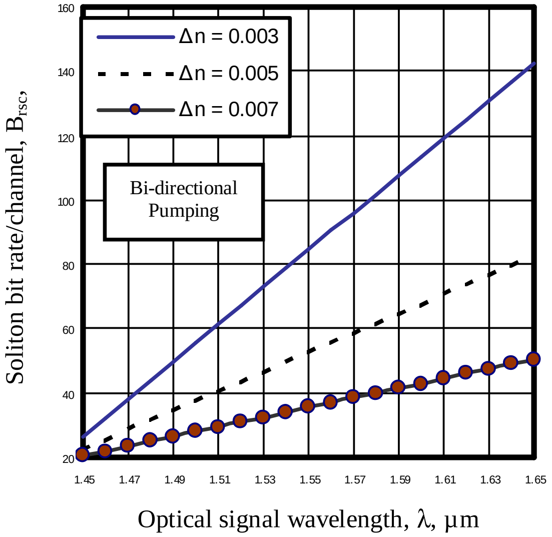


Figure 4. Variations of the soliton bit rate per channel with optical signal wavelength at the assumed set of parameters.

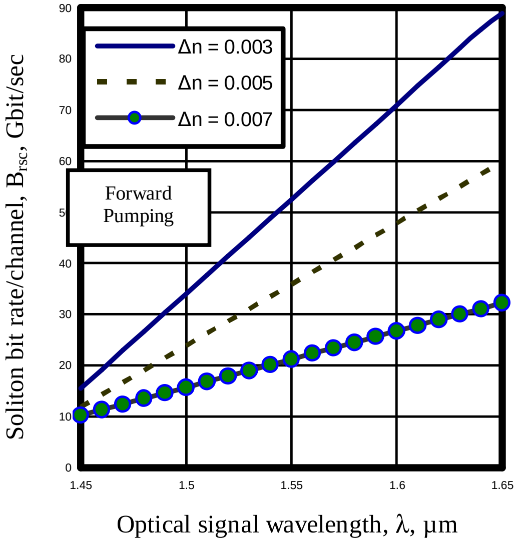


Figure 5. Variations of the soliton bit rate per channel with optical signal wavelength at the assumed set of parameters.

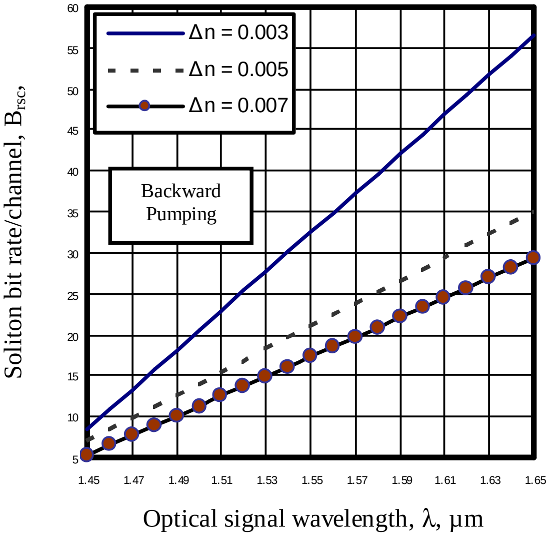


Figure 6. Variations of the soliton bit rate per channel with optical signal wavelength at the assumed set of parameters.

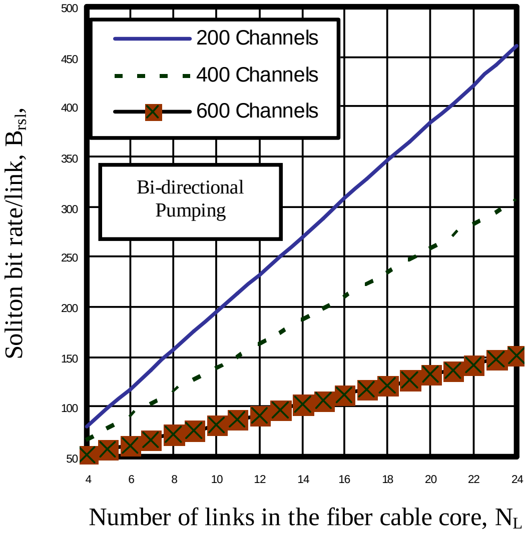


Figure 7. Variations of soliton bit rate per link with number of links in the fiber cable core at assumed set of parameters.

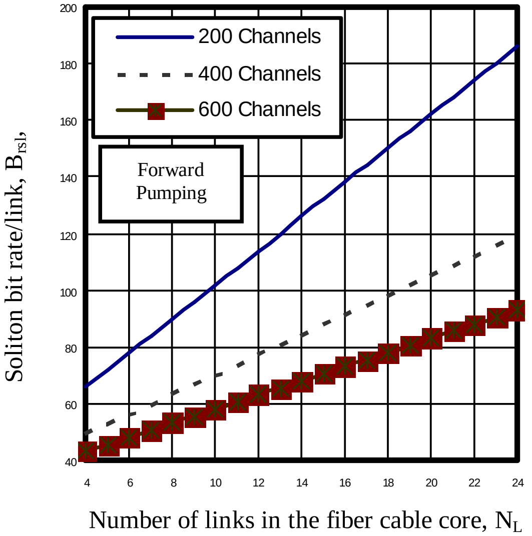


Figure 8. Variations of soliton bit rate per link with number of links in the fiber cable core at assumed set of parameters.

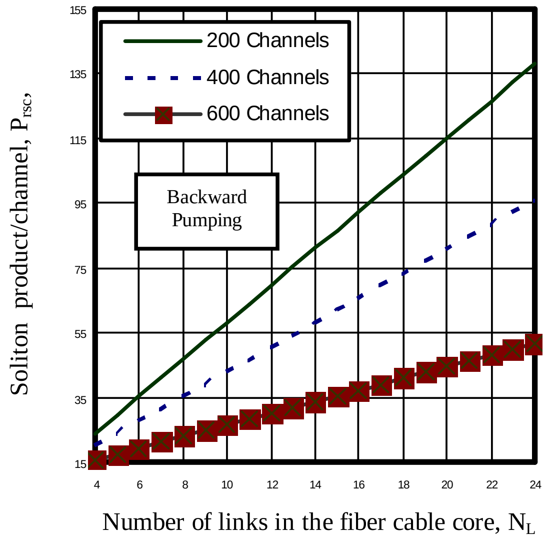


Figure 9. Variations of soliton bit rate per link with number of links in the fiber cable core at assumed set of parameters.

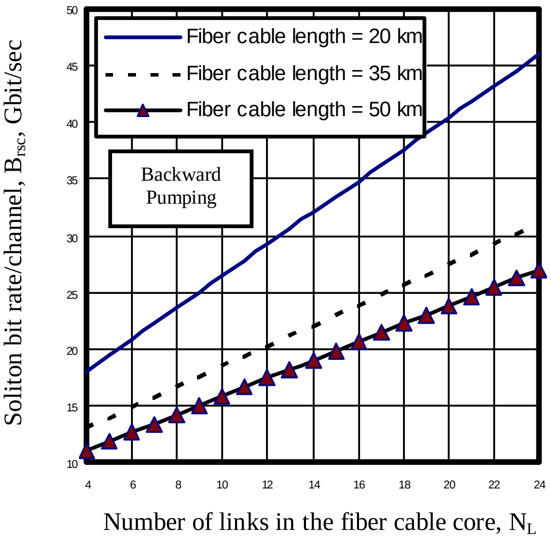


Figure 12. Variations of soliton bit rate per channel with number of links in cable at the assumed set of parameters.

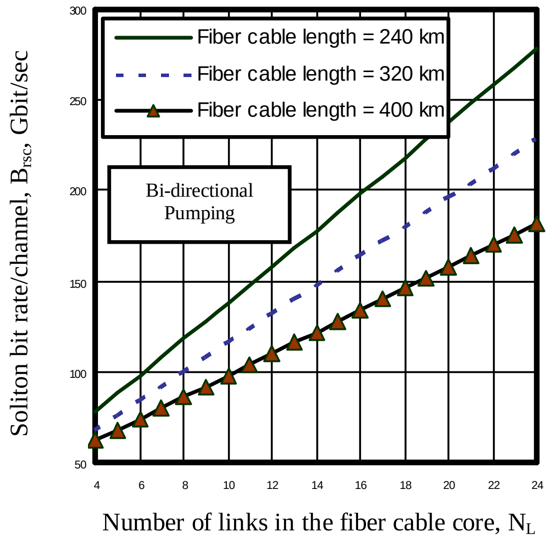


Figure 10. Variations of soliton bit rate per channel with number of links in the cable at assumed set of parameters.

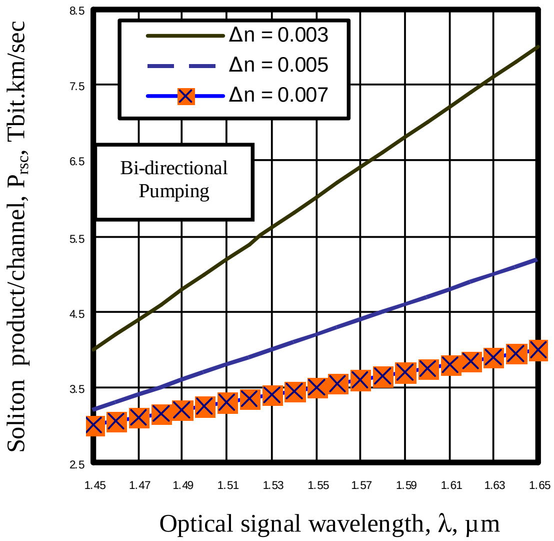


Figure 13. Variations of the soliton product per channel with optical signal wavelength at the assumed set of parameters.

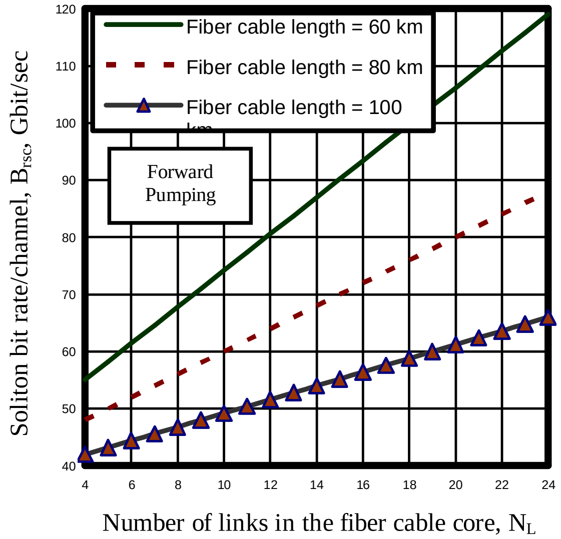


Figure 11. Variations of soliton bit rate per channel with number of links in cable core at assumed set of parameters.

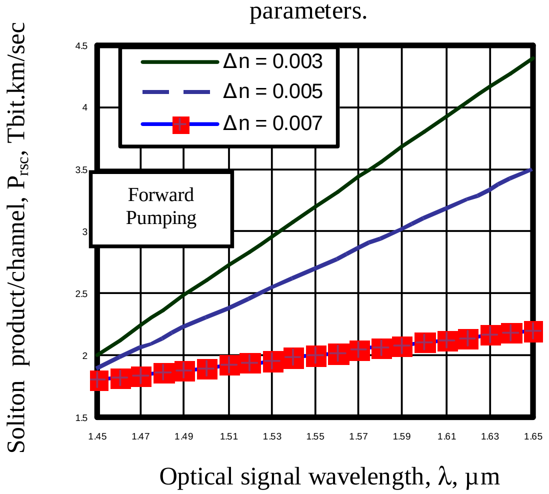


Figure 14. Variations of the soliton product per channel with optical signal wavelength at the assumed set of parameters.

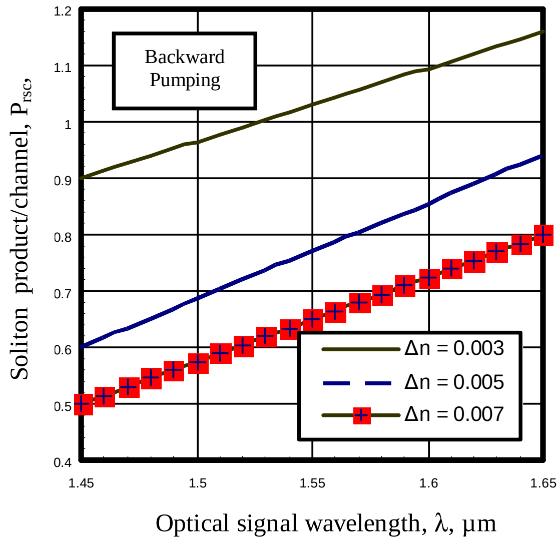


Figure 15. Variations of the soliton product per channel with optical signal wavelength at the assumed set of parameters.

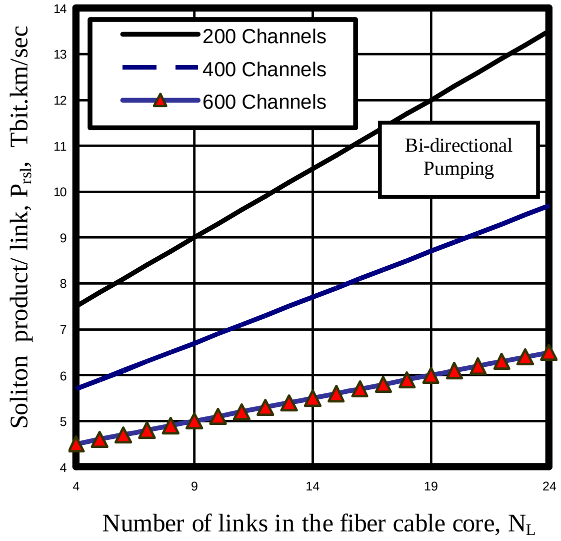


Figure 16. Variations of soliton product/link with number of links in cable core at the assumed set of parameters.

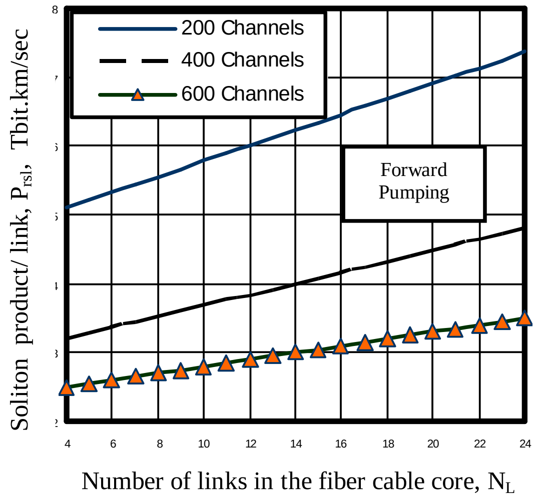


Figure 17. Variations of soliton product/link with number of links in the cable core at the assumed set of parameters.

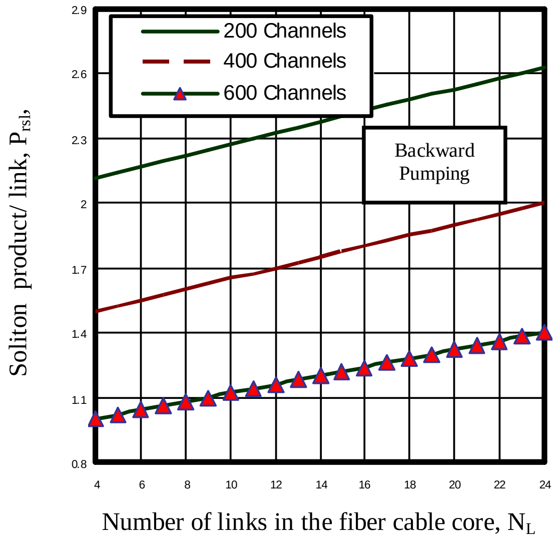


Figure 18. Variations of soliton product/link with number of links in fiber cable core at the assumed set of parameters.

Based on the set of Figs (3-18), the following facts and obtained features are assured to present the high performance of fiber Raman amplifiers in modern optical telecommunication networks to give the best transmission bit rates and distances for different pumping direction configurations as follows:

1) Figure 3 has assured that as the on-off Raman gain increases, the NF decreases for different fiber cable lengths. Here with the fiber cable length equal to 150 km presents the highest NF than the two other fibber cable lengths at the same on-off Raman gain.
2) As shown in Figs. (4-6), we have demonstrated that as the optical signal wavelength increases, the soliton bit rate per channel also increases at the same relative refractive-index difference Δn. But as Δn increases, soliton bit rate per channel decreases at the same optical signal wavelength for different pumping direction configurations. Moreover, in the case of bi-directional pumping configuration presents the highest bit rates per channel than the other two pumping direction configurations.
3) Figs. (7-9) have proved that as the number of links in the fiber cable core increases, soliton bit rate per link also increases at the same number of transmitted channels, and as the number of transmitted channels increases, the soliton bit rate per link decreases at the same number of links in the fiber cable core. But in the case of bi-direction pumping configuration presents the highest bit rates per link than the other two pumping direction configurations.
4) As shown in Figs. (10-12), we have indicated that as the number of links in the fiber cable core increases, the soliton bit rate per channel also increases at the same fiber cable lengths, and as the fiber cable length increases, the soliton bit rate per channel decreases at the same number of links in the fiber cable core. Moreover, in the case of bi-direction pumping configuration, presents the highest transmission bit rates per channel and the highest transmission distances than the other two pumping direction configurations.

5) The series of Figs. (13-15) have assured that as the optical signal wavelength increases, the soliton product per channel also increases at the same relative refractive-index difference Δn. But as Δn increases, soliton product per channel decreases at the same optical signal wavelength for different pumping direction configurations. But in the case of bi-directional pumping configuration, presents the highest soliton product per channel than the other two pumping direction configurations.
6) As shown in the series of Figs. (16-18) have demonstrated that as the number of links in the fiber cable core increases, soliton product per link also increases at the same number of transmitted channels, but as the number of transmitted channels increases, the soliton product per link decreases at the same number of links in the fiber cable core. Moreover, in the case of bi-direction pumping configuration presents the highest soliton product per link than the other two pumping direction configurations.

6. Conclusions

In a summary, we have investigated and analyzed numerically and parametrically the ultra high speed of the distributed optical fiber Raman amplifiers in modern DWDM optical fiber telecommunication networks over wide range of the affecting and controlling parameters. We have demonstrated in our major interest the importance role of distributed fiber Raman amplifiers to strength the optical signal power in the forward, backward, and bi-directional pumping direction configurations and thus allowing longer transmission distances, higher capacity, and maximum transmission bit rates either per link or per channel. But especially in the bi-direction pumping configuration has presented the best and highest performance either in transmission distances or transmission bit rates. It is evident that the higher of on-off Raman gain, the lower of noise figure along total fiber cable length that is achieved in forward, and bi-directional pumping direction configurations. Moreover, we have assured that in the bi-directional pumping configuration has presented the highest soliton transmission bit rates and products either per link or per channel, and the highest transmission distances than the other two pumping direction configurations.

References

[1] H. Masuda, A. Mori, K. Shikano, and M. Shimizu, “Design and Spectral Characteristics of Gain-Flattened Tellurite-Based fiber Raman amplifiers,” J. Lightw. Technol., Vol. 24, No. 1, pp. 504-515, Jan. 2006.

[2] K. Morito, “Output-Level Control of Semiconductor Optical Amplifier by External Light Injection,” J. Lightw. Technol., Vol. 23, No. 12, pp. 4332-4341, Dec. 2005.

[3] P. Velanas, A. Bogris, and D. Syridis, “Impact of Dispersion Fluctuations on the Noise Properties of Fiber Optic Parametric Amplifiers,” J. Lightw. Technol., Vol. 24, No. 5, pp. 2171-2178, May 2006.

[4] P. Kylemark, H. Sunnerud, M. Karlsson, and P. Andrekson “ Semi-Analytic Saturation Theory of Fiber Optic Parametric Amplifiers,” J. Lightw. Technol., Vol. 24, No. 9, pp. 3471-3479, Sept. 2006.

[5] F. Yaman, Q. Lin, S. Radic, and G. Agrawal “Fiber-Optic Parametric Amplifiers in the Presence of Polarization Mode Dispersion and Polarization-Dependent Loss,” J. Lightw. Technol., Vol. 24, No. 8, pp. 3088-3096, Aug. 2006.

[6] M. Schneiders, S. Vorbeck, R. Leppla, E. Lach, M. Schmidt, S. Papernyi, and K. Sanapi “Field Transmission of 8×170 Gb/s Over High-Loss SSMF Link Using Third Order Distributed Raman amplification,” J. Lightw. Technol., Vol. 24, No. 1, pp. 175-190, Jan. 2006.

[7] Mahmud Wasfi, “Optical Fiber Amplifiers Review,” International Journal of Communication Networks and Information Security (IJCNIS), Vol. 1, No. 1, pp. 42-47, April 2009.

[8] G. Felinsky, and P. Korotkov, “Raman Threshold and Optical Gain Bandwidth in Silica Fibers,” Journal of Semiconductor Physics, Quantum Electronics, and Optoelectronics, Vol. 11, No. 4, pp. 360-363, July 2008.

[9] B. G. Lee, A. Biberman, A. C. Tuner, M. A. Foster, M. Lipson, A. L. Gaeta, and K. Bergman, “Demonstration of Broadband Wavelength Conversion at 40 Gb/sec in Silicon Waveguides,” IEEE Photonics Technology Letters, Vol. 21, No. 3, pp. 182-184, Feb. 2009.

[10] L. T. Jordanova, and V. I. Topchiev, “Improvement of the Optical Channel Noise Characteristics Using Distributed Raman Amplifiers,” ICEST, Vol. 12, No. 5, pp. 20-23, June 2008.

[11] S. Raghuawansh, V. Guta, V. Denesh, and S. Talabattula, “Bi-directional Optical Fiber Transmission Scheme Through Raman Amplification: Effect of Pump Depletion,” Journal of Indian Institute of Science, Vol. 5, No. 2, pp. 655-665, Dec. 2006.

[12] W. Fleming, “Dispersion in GeO_2-SiO_2 Glasses,” Applied Optics, Vol. 23, No. 24, pp. 4486-4493, 1985.

[13] G. Yabre, “Theoretical Investigation on the Dispersion of Graded-Index Polymer Optical Fiber,” Journal of Lightw. Technol., Vol. 18, No. 16, pp. 869-882, 2000.

[14] Abd El-Naser A. Mohamed, Abd El-Fattah Saad, and Ahmed Nabih Zaki Rashed, “High Channel Arrayed Waveguide Grating (AWG) in Wavelength Division Multiplexing Passive Optical Networks (WDM-PONs),” IJCSNS International Journal of Computer Science and Network Security, Vol. 9, No. 1, pp. 253-259, Jan. 2009.

Using a Novel Intelligent Location Management Strategy in Wireless Networks

J. Amar Pratap Singh[1], Marcus Karnan[2]

[1] Research Scholar
Department of Computer Science and Engineering
Anna University, Coimbatore
japsindia@yahoo.com

[2] Prof & Head
Department Computer Science and Engineering
TamilNadu College of Enineering, Coimbatore
karnanme@yahoo.com

Abstract: ***Recent advances in cellular mobile systems provides access to a wide range of services and allow mobile terminals (MTs) to move randomly from one place to another within a well-defined geographical area. Due to the growing number of mobile users, global connectivity, and the small size of cells, one of the most critical issues regarding these networks is location management. The challenging task in a cellular system is to track the location of the mobile users effectively so that the connection establishment cost and delay is low. In recent years, several strategies have been proposed to improve the performance of the location management procedure in cellular networks. In this paper, we propose an intelligent approach by taking the User Profile History (UPH); to reduce the location update cost by combining Back-Propagation Algorithm and Cascaded Correlation Neural Network. The implementation of this strategy has been subject to extensive tests. The results obtained confirm the efficiency of UPH in significantly reducing the costs of both location updates and call delivery procedures when compared to the various other strategies well-known in the literature.***

Keywords: Location Update, User Profile History, Mobile Terminal, Home Location Register, Location Management, Back-Propagation, Neural Networks.

1. Introduction

Owing to the increasing population of mobile subscribers, smaller sized cells have been used to accommodate the large number of mobile terminals (MT's). The location management in a cellular network is used to track the location of the user. Location management has two operations [15]: (a) location update: tracking information is carried out by the location update procedure at the location area and (b) paging: when a call arrives, paging procedure is used to find out the location of the target user. Sending paging signals to all cells within a location area (LA) to locate an MT may result in an excessive amount of network bandwidth. Therefore, more sophisticated schemes [1,2] were proposed to make the location update and terminal paging operations more efficient. These schemes include the time-based, movement-based and distance-based schemes which locate an MT by paging the LA's ring by ring from its last updated location. In this paper, we propose a location update scheme based on cascaded correlation neural networks where an MT updates its location only when it's moving direction changes. To locate an MT, paging can be carried out along its moving direction, and hence the paging cost is reduced. Moreover the MT's moving direction can be determined by simple numerical calculations.

Location management methods are classified into two major groups: Memory-based and non-memory-based methods. The first group includes methods based on learning processes, which require knowledge of mobile user behavior, while the second group includes methods based on specific algorithms and network architectures. The strategy proposed in this paper belongs to the first group. In North America, the IS-41 standard is used for both the location update and call delivery procedures. This standard deploys a two-level database architecture consisting of a single home location register (HLR) and several visitor location registers (VLR). The HLR for a given network contains the network's subscriber profiles, while a VLR stores the profiles of the users that are currently roaming within LAs associated with that specific VLR. Third-generation mobile networks are characterized by high user density and high mobility (like current 2G systems) and small cell sizes, which will increase the number of location updates and handoff messages, thus limiting the switching capacity and available bandwidth. Reducing the signaling and database access costs of location management introduces significant technical challenges which have to be dealt with and constitutes an important research area. Several alternative strategies have recently been proposed to improve the performance of the location management scheme [3, 4, 5, 6, 7].

There are two basic operations in location management: location update and paging. Location update is the process through which system tracks the location of mobile terminals that are not in conversations. The mobile terminal reports its up-to-date location information dynamically. A PA may include one or more cells. When an incoming call arrives, the system searches for the mobile terminal by sending polling signals to cells in the PA. This searching process is referred to as paging. To perform location update or paging will incur a significant amount of cost (e.g., wireless bandwidth and processing power at the mobile terminals, the base stations, and databases), which should be minimized in the systems.

A user profile History (UPH) associates with each user a list of location areas (LA) where she is most likely to be located within a given time interval. When a call arrives for an MT, each location within the list is paged sequentially until the MT is found. When a user moves between locations within the list, no location update is required. The list is stored at an intermediate location database (ILD) associated with a Mobile Switching Centers (MSC) as well as within the user's MT. The cost reduction depends on the behavior of each class of user. It can be assumed that, when the user follows its expected behavior, the cost of a location update is reduced.

In the UPH, if the position of a user is always known in advance, then no explicit registration is necessary. Thus, the optimal location area is given by a single cell, which, in turn, minimizes paging costs. Stationary users on fixed schedules exhibit this type of behavior. If mobile users are classified into categories, as was done earlier, the system could treat each category differently to minimize system costs. Furthermore, user mobility information can be used to assist mobility management (traffic routing), to manage network resources (resource allocation, call admission control, congestion, and flow control), and to analyze handoff algorithms in integrated wired/wireless networks. User mobility patterns can also be useful for system recovery [8]. By means of a fuzzy logic algorithm, a users' location is forecast by the system, which eliminates the need of a backup. This way, users will not experience long service delays.

Our strategy differs from the others user profile strategies in the following aspects:

We use a cascaded correlation artificial neural network (CCANN) [9] to learn about the users' regular routines. Pattern recognition is one of the fields where neural networks (ANNs) have been strongly applied. Pattern learning and classification can be stated as the problem of labeling test patterns derived by a particular application domain. A classification system may be trained by a set of data features, adequately prepared or not. We have divided our strategy into two steps: training and application. In general, ANN systems are capable of "learning" trends in a given data set and establishing input-output relationships based strictly on a "test" set of data. It is desirable for the "test" data that the system "learns" from to be as representative of the complete data set as possible; trends not seen in the test data set will not be "learned" by the neural network system. After training, the network is ready for application. For satisfactory application, it is essential that the training data contain input sets (and the associated output values) that represent the entire range of possible future inputs; the system will only perform as well as it has been trained. The training examples may contain errors. Neural Network (NN) learning methods are quite robust to noise in the training data. The ability to learn is a fundamental trait of intelligence. Although a precise definition of learning is difficult to formulate, a learning process in the NN context can be viewed as the problem of updating network architecture and connection weights (an elementary structure and functional unit between two neurons) so that a network can efficiently perform a specific task. The network usually must learn the connection weights from available training patterns. Performance is improved over time by iteratively updating the weights in the network. NN's ability to automatically learn from examples makes them attractive. Instead of following a set of rules specified by human experts, NNs appear to learn underlying rules (like input-output relationships) from the given collection of representative examples. This is one of the major advantages of neural networks over traditional expert systems. Neural networks derive their computing power through their ability to learn and then generalize; generalization refers to the ability of the neural network to produce reasonable outputs for inputs not encountered during training. It is this quality that we utilize to predict the movement of mobile users so that we can predict the position of a user in advance and reduce the paging cost based on the predicted destination cell.

Finally, the impact on the performance of location management with CCANN is reduced. The cost of the UPL is decomposed into four components: training procedure, maintenance and update of the user's profile, location update, and call delivery. Although CCANN learning times are relatively long, evaluating the learned network in order to apply it to a subsequent instance (maintenance and update of the user's profile, location update and call delivery) is very fast. In CCANN, performance is improved over time by iteratively updating the weights in the network as compared to conventional ANN algorithms. This paper proposes a user pattern learning strategy that reduces the signaling cost of a location update by increasing the intelligence in the location procedure.

2. System Model and Mobility Management

A system Model based on cellular networks is described, and also mobility management is presented below.

2.1 System Model

Consider a mobile wireless network with a cellular infrastructure e.g Universal Mobile Telecommunications Service (UMTS) which can be viewed as an evolution of GSM that supports 3G services. Generally, a UMTS network is divided in an access network and a core network (Fig. 1). The former is dependent on the access technology, while the latter can theoretically handle different access networks. The access network is known as the UMTS Terrestrial Radio Access Network (UTRAN). The UTRAN is comprised of two types of nodes, the Radio Network Controller (RNC) and the Node B, which is a base station (BS). It controls the radio resources within the network and can interface with one or more stations (Node Bs).Uu in UMTS. The UTRAN communicates with the core network over the Iu interface. The Iu interface has two components: the Iu-CS interface, supporting circuit-switched (CS) services, and the Iu-PS interface, for packet-switched (PS) services. The air interface used between the user equipment (UE) and the UTRAN is WCDMA. This interface is called The RNC that controls a given Node B is known as the Controlling RNC (CRNC). For a given connection between a UE and the core network, only one RNC can be in control, the Serving RNC (SRNC). The CRNC controls the management of radio resources for the Node B that it supports. The SRNC controls the radio resources that the UE is using. It is possible for a CRNC and a SRNC to coincide. UTRAN supports soft handovers (where an UE is communicating with a Node B whose CRNC is not the SRNC). The Iur interface's purpose is to support this type of handover, that is, inter-RNC mobility.

Mobility management issues in multitier PCS systems are presented in [3]. Hwang et al. [10] propose a direction-based location update method that uses line paging for reducing paging costs. This approach is based on a conventional two-dimensional random walk model, where the directions of the MTs are assumed to be independent and identically distributed.

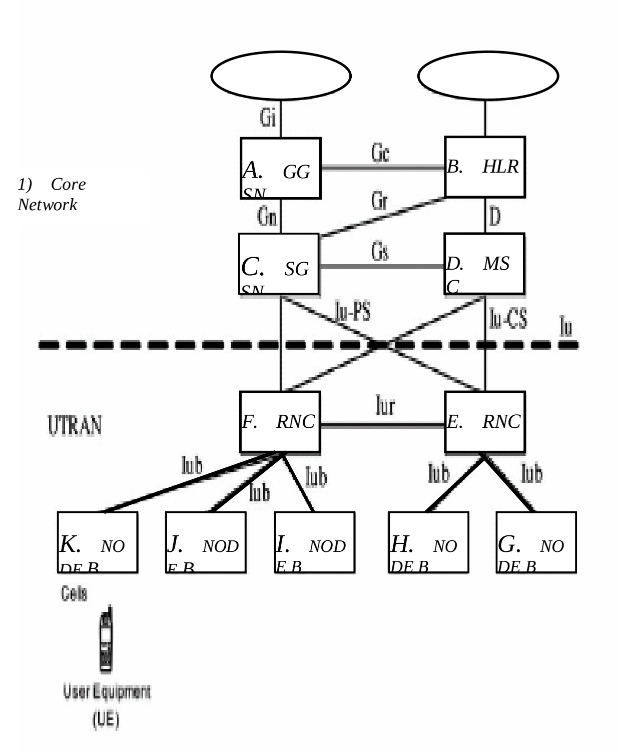


Figure 1. UMTS network architecture

Aljadhai and Znati [11] deal with the question of how to support Quality of Service (QoS) through an exact knowledge of the trajectory followed by the MT. For this solution to work, it is crucial to have an efficient prediction mechanism for the user's trajectory. With an estimation of trajectories, arrival, and departure times, the system is able to support QoS requirements. This approach does not necessitate the memorization of users' profile within the network.

Wu et al. [12] present a new analytic framework for dynamic location management of PCS networks.

2.2 Mobility Management

In standard UMTS, Mobile Switching Centers (MSCs) are responsible for the circuit switched location management, while Serving GPRS Support Nodes (SGSNs) assume the packet switched location management. Both domains are linked over some interfaces, but the information is kept in separate network entities: The CS location information is in the MSC, while the PS location information is in the SGSN. The HLR is a common location information database for both domains. Several area types have been defined in UMTS to handle the location information:

- Location Areas (LA) are the same as in GSM.
- A Routing Area (RA) is composed of a group of cells that belong to only one RA. Several RAs can be included in the same LA, but an RA cannot span more than one LA. The PS domain to track the MT's location when in idle mode uses the RAs.
- UMTS Registration Areas (URAS) are an intermediate level between cells and RAs (or LAs). They are similar to RAs and LAs, but are used by the UTRAN to set trade-offs between the MT's location accuracy and signaling load. Furthermore, they are used to track the MT's location while it is in connected mode without using a logical channel. This concept is optional in UTRAN.
- Cells are related to the provision of radio coverage. The idea of having this diversity is to allow a trade-off between location accuracy and paging [3].

3. User Profile History

In this section, we introduce our UPH location management strategy. We also present protocols and algorithms supporting this scheme. Furthermore, several application scenarios are proposed. In the following discussion, we define the anchor LA of an MT as the LA whose location is updated at the HLR. When an MT changes LA, it updates the pointer at the anchor LA. This way, no update is made at the HLR.

3.1 Overview of the User Profile History

The main motivation behind the use of artificial neural networks (ANN) is their ability to learn relationships in complex data sets that may not be easily perceived by humans. Learning and generalization are perhaps the most important topics in neural network research. Learning is the ability to approximate the underlying behavior adaptively from the training data, while generalization is the ability to predict well beyond the training data. The basic element of an artificial neural network (ANN) system is called a neuron. The neuron accepts one input x, which may actually be a sum of multiple inputs, and produces an output value y based on a nonlinear function. In general, ANN systems are capable of "learning" trends in a given data set and establishing input-output relationships based strictly on a "test" set of data. It is desirable for the "test" data that the system "learns" from to be as representative of the complete data set as possible; trends not seen in the test data set will not be "learned" by the neural network system [13].

Typically, ANNs are organized in several layers of elementary units called neurons, each computing a non-linear function of a weighted sum of its inputs. The learning phase of an ANN is based on algorithms (e.g., backpropagation) able to set the weight connections by training the net with a known data set until a certain (small) error is achieved. Unfortunately, the design of networks of practical interest is not an easy task as the whole topology of the network, i.e., number of hidden layers, number of neurons inside each hidden layer, the connections between the neurons, etc., should be specified in advance. The hidden layers are called "hidden" units because their output is available only within the network and is not available as part of the global network output. Typically, trial-and-error and heuristic procedures are used to determine the network topology according to some criteria (e.g., minimum classification error). In practice, for a given network topology and (real) training set, the best possible set of weights cannot be guaranteed by the back propagation algorithm [13].

The number of nodes used in the input layer usually depends on the type and amount of input data; several hundred input nodes may be used in large applications. The number of nodes in the hidden layer determines, in general, the ability of the network to learn complex relationships. There may be multiple hidden layers to increase the network's ability to learn. In our model, with the BP algorithm, there are three layers in the Neural Networks, input layer, hidden layer, and output layer. The role of the hidden layer is to remap the inputs and results of previous layers to achieve a more separable or classifiable representation of the data and allow attachment of semantics to certain combinations of layer inputs.

ANNs perform their calculations using nonlinear functions and simple multiplying factors, called weights, which are associated with a pathway between any two nodes. While the functions remain constant for any given application, the weights are updated in such a manner that the complete network "learns" to produce a specific output for a specific input. The process of adjusting the weights to achieve a specified accuracy level is referred to as "training." The backpropagation (BP) training algorithm is a method for iteratively adjusting the weighting factors until the desired accuracy level is achieved. This algorithm is based on a gradient-search optimization method applied to an error function (i.e., the sum of squared error). In our approach, we use the learning process to derive a list from which we can find, with high accuracy, the exact cell in which the MT resides at any time of every day. The learning process is able to derive such a list after observing (learning) the behavior of a mobile user for a certain period of time.

By observing the mobile user's daily behavior, we use the BP algorithm to learn the behavior. With some useful data from observation of the mobile user as the input nodes, we can obtain the output as the result we want, which is the cell information of the mobile user on observation, that is to say, the cell list for a mobile user.

For every mobile user there is a user pattern learning process associated to it. We may classify the users into three different categories depending the predictability of their daily routine: users who have a very high probability of being where the system expects them to be (deterministic users), users who have a certain likelihood of being where the system expects them to be (quasi-deterministic users), and users whose position at a given moment is unpredictable (random users), similar to the classification proposed in [14]. The predictability of deterministic and quasi-deterministic users can be used by the system to reduce the number of location update operations. So, after the learning process completes, we get the mobile user's behavior associated with known location areas. Then, a profile is built for the mobile user (Fig. 3). When a call arrives for a mobile, it is paged sequentially in each location within the list. When a user moves between location areas in this list, no location updates are required. The list is stored at the HLR in the information database (ID) as well as in the user's mobile terminal. The cost reduction depends on the behavior of each class of user. It can be assumed that, when the user follows its expected behavior, the location update cost is reduced, even if accesses to HLR are minimized when calls are received from relatively close areas.

Our strategy increases the intelligence of the location update procedure and utilizes replication and locality to reduce the cost incurred from the paging procedure.

An Intermediate Location Database (ILD) is added to the UPL scheme. This database is located on the same architectural level as the MSC and contains the profile of each user. Furthermore, the ILD contains a flag for each registered MT and indicates whether or not the MT is roaming under that particular MSC. The flag helps us exploit the "locality" property of calls (i.e., incoming calls from the same MSC). Moreover, our strategy takes advantage of calls placed regularly to a particular user (i.e., a great amount of calls are placed by the MT's top five callers). Thus, some ILDs will store "location data tables" that contain pointers tracking the called MT's location. This may somewhat increase the location update cost since the MT must use these pointers each time it changes location. On the other hand, using pointers significantly reduces paging costs.

An user behaviour is as shown in Fig.3. A user's profile is made up of several fields. The first field is the profile number. Each user might have several profiles, each containing a different behavioral pattern. The next two fields contain the IDs of the MSC and the LA under which an MT might be roaming. The Expected Entry Time (EET) field indicates the time interval within which the system expects to locate the MT in a particular LA. The Timestamp field indicates the date and time an MT has entered a new LA. Finally, the Number of Visits field saves the number of times that an MT has been roaming under a particular LA. When a MT enters a new LSTP that he has never seen before, the MSC creates a basic profile containing a single record with MT's current LA.

The UMTS network consists of three interacting domains: Core Network, UMTS Terrestrial Radio Access Network, and Mobile Terminal (Fig.1). The main function of the core network is to provide switching, routing, and transit for user traffic. The Core network also contains databases (like HLR) and network management functions. The UPH process is in the Mobile Switching Center (MSC) in the core network, The UPH associates with each user a list of location areas (LA) where she is most likely to be located within a given time interval. This list is stored at an intermediate location database (ILD) associated with an MSC as well as within the user's MT.

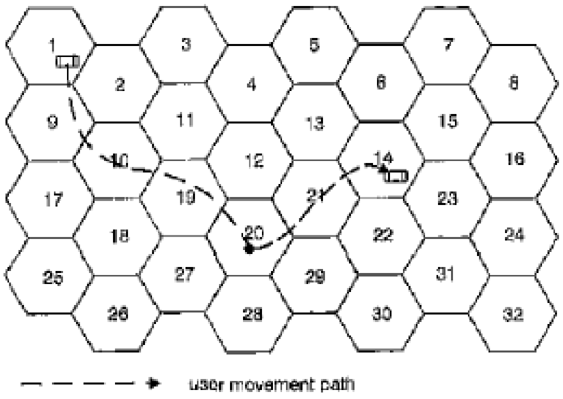


Figure 2. User Movement Path

3.1.1 Backpropagation Algorithm

We use the BP algorithm to implement the learning process. In our problem, the learning process aims to derive a list with which we can find the cell in which the MT locates at any time of every day with high accuracy after observing the behavior of the mobile user for a period, for example, a month.

There are three layers in the Neural Networks: input layer, hidden layer, and output layer. The role of the hidden layer is to remap the inputs and results of previous layers to achieve a more separable or classifiable representation of the data and allow attachment of semantics to certain combinations of layer inputs. In this case, there are three layers, the input layer that contains the values for Day and Time, a hidden layer that contains three nodes, h1, h2, h3, and two output units that gives the value of the output values, it's the probability that reside each cell in the user mobile history (UMH). The hidden layer is so named because the network can be regarded as a black box with inputs and outputs that can be seen, but the hidden units cannot be seen.

To compute the value of the output units, residence probability, we place values for Day and Time on the input layer units. Let these values be Monday, 10:15, as shown in Table 1. First, we compute the value of the hidden layer unit, hs. The first step of this computation is to look at each lower level unit. For each of these connections, find the value of the unit and multiply by the weight and sum all the results.

Table 1: An Example Training Set.

Examples	**Di(Day)**	**Ti(Time)**	**Ci (Cell Id)**	**Probability (%)**
E1	Monday	02.15	5,1,6	90,50,10
E2	Monday	11.45	5,2,1	95,50,10
E3	Sunday	12.00	5,6,3	40,90,15
E4	Thursday	14.15	5,2,1	95,50,10
E5	Thursday	05.30	5,1,6	90,50,10

3.1.2 The Steps of BP

The error back – propagation algorithm can be outlined as
Step 1: Initialize all weights to small random values.
Step 2: Choose an input-output training pair.
Step 3: Calculate the actual output from each neuron in a layer by propagating the signal forward through the network layer by layer (forward propagation).
Step 4: Compute the error value and error signals for output layer.
Step 5: Propagate the errors back ward to update the weights and compute the error signals for the preceding layers.
Step 6: Check whether the whole set of training data has been cycled once, yes – go to step 7; otherwise go to step 2.
Step 7: Check whether the current total error is acceptable; yes- terminate the training process and output the field weights, otherwise initiate a new training epoch by going to step 2.

3.1.3 The Cascade Correlation Algorithm

1. Initial configuration: The algorithm begins with a simple perceptron with N input units [9] and M output units. N and M are chosen on the basis of the problem that the network is to learn as shown in Fig.4.

2. Initial training: The perceptron is trained on the entire training set {(Vp,Tp) | p = 1, . . . , P }, until the performance of the network is as good as possible. If the desired performance is obtained, the algorithm stops. Otherwise: Start adding hidden units to the network, one by one.

3. Training of candidates: A pool of candidates for a new hidden unit is generated. This pool emulates a stochastic search in the weight space, which will decrease the risk of inserting a candidate stranded in a local minimum with high error. Each node in the pool of candidates is connected to all input nodes and all previously inserted hidden units. Each of the candidates is trained with the purpose of maximizing some measure of ”goodness” of the candidate.

4. Inserting a new hidden unit: The candidate with the highest score is inserted “for real” in the network as a new hidden unit. The incoming weights to the new hidden unit are then frozen, i.e. they are not to be changed anymore. The new hidden unit is connected to all output nodes with random weights.

5. Retraining the network: All the incoming weights to the output units are retrained in order to adjust the weights from the newly inserted hidden unit. If the performance of the network is satisfying after retraining, the algorithm stops. Otherwise: Go to 3.

3.1.4 Cascade Correlation Neural Network Architecture

A cascade correlation network consists of input units, hidden units, and output units. Input units are connected directly to output units with adjustable weighted connections. Connections from inputs to a hidden unit are trained when the hidden unit is added to the net and are then frozen. Connections from the hidden units to the output units are adjustable consequently.

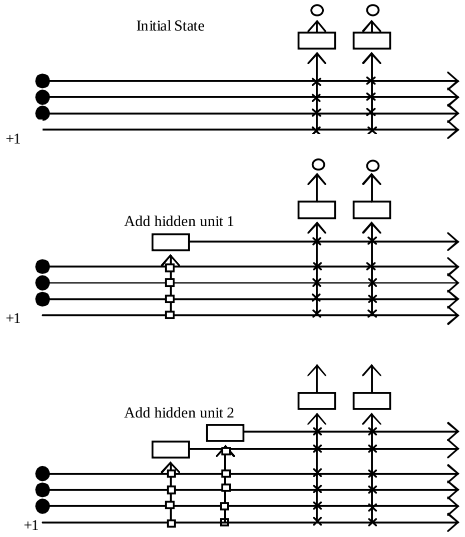


Figure 3. The Cascade architecture, initial state and after adding two hidden units.

Cascade correlation network starts with a minimal topology, consisting only of the required input and output units (and a bias input that is always equals to 1). This net is trained until no further improvement is obtained. The error for each output until is then computed (summed over all training patterns). Next, one hidden unit is added to the net in a two-step process. During the first step, a candidate unit is connected to each of the input units, but is not connected to the output units. The weights on the connections from the input units to the candidate unit are adjusted to maximize the correlation between the candidate's output and the residual error at the output units. The residual error is the difference between the target and the computed output, multiplied by the derivative of the output unit's activation function, i.e., the quantity that would be propagated back from the output units in the back propagation algorithm. When this training is completed, the weights are frozen and the candidate unit becomes a hidden unit in the net. The second step in which the new unit is added to the net now begins. The new hidden unit is then connected to the output units, and the weights on the connections being adjustable. Now all connections to the output units are trained. (Here the connections from the input units are trained again, and the new connections from the hidden unit are trained for the first time.) A second hidden unit is then added using the same process. However, this unit receives an input signal from the both input units and the previous hidden unit. All weights on these connections are adjusted and then frozen. The connections to the output units are then established and trained. The process of adding a new unit, training its weights from the input units and the previously added hidden units, and then freezing the weights, followed by training all connections to the output units, is continued until the error reaches an acceptable level or the maximum number of epochs (or hidden units) is reached.

The purpose of inserting a new unit is to reduce the total error of the network. The way the CCA does this is to train the candidate unit so the correlation between the residual error and the output from the candidate is maximized. Let X and Y be two stochastic variables.

Then the correlation between X and Y is defined as:

$$Corr[X,Y] = \frac{Cov[X,Y]}{\sqrt{Var[X]Var[Y]}}$$

4. Implementation and Analysis

We analyze the sensitivity of the CCNN algorithm to the learning rate in the UPH approach. We generated 2400 samples for each set, assuming normal distributions using the statistics estimated from the real data. We used 1500 samples of each set as training data and the rest as test data. Then, we chose to use the generated data to avoid the problem of incorrect estimation of parameters. If training data is not a good representative of the sets, then the classification accuracies of training and test data might be very different, resulting in difficulty in comparing the performance of the learning algorithms. The training performance and the classification accuracies are shown in Fig.5 and Table 2 respectively.

Table 2: Classification Accuracies of CC Neural Networks

Learning Rate	Training Data	Test Data
0.01	86.124	85.2
0.3	85.202	84.72

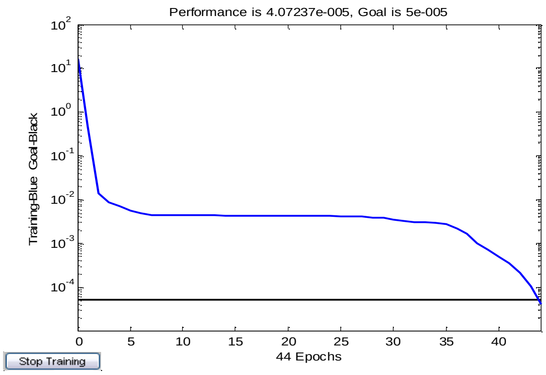


Figure 4. Training Performance Curve

5. Conclusions

In this paper, we present a user Profile History strategy using back propagation and cascaded neural networks to reduce location update signaling cost by increasing the intelligence of the location procedure in UMTS. This strategy associates to each user a list of cells where she is likely to be with a given probability in each time interval. The list is ranked from the most likely to the least likely place where a user may be found. When a call arrives for a mobile, it is paged sequentially in each location within the list. When a user moves between location areas in the list, no location updates are required. The results obtained from our performance evaluation confirm the efficiency and the effectiveness of UPH in comparison with the UMTS standard and other well-known strategy. This improvement represents a large reduction in location update and paging signaling costs.

References

[1] F. Akyildiz, J. S. M. Ho, and Y.-B. Lin, “Movement-based location update and selective paging for PCS networks,” IEEE/ACM Trans. Networking, vol. 4, pp. 629–638, Aug. 1996.

[2] D. G. Jeong and W. S. Jeong, “Probabilistic location update for advanced cellular mobile networks,” IEEE Commun. Lett., vol. 2, pp. 8–10, Jan. 1998.

[3] E. Cayirci and I.F. Akyildiz, "User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems," IEEE Trans. Mobile Computing, vol. 1, no. 3, pp. 236-247, July-Sept. 2002.

[4] R. Chen and B. Gu, "Quantitative Analysis of a Hybrid Replication with Forwarding Strategy for Efficient and Uniform Location Management in Mobile Wireless Networks," IEEE Trans. Mobile Computing, vol. 2, no. 1, pp. 3-15, Jan.-Mar. 2003.

[5] J. Ho and I. Akyildiz, "Dynamic Hierarchical Database Architecture for Location Management in PCS Networks," IEEE/ACM Trans. Networking, vol. 5, no. 5, pp. 646-660, 1997.

[6] J. Ho and I. Akyildiz, "Mobile User Location Update and Paging under Delay Constraints," ACM Wireless Networks, vol. 1, no. 4, pp. 413-425, 1995.

[7] W. Hu and T. Tan, "A Hierarchical Self-Organizing Approach for Learning the Patterns of Motion Trajectories," IEEE Trans. Neural Networks, vol. 15, no. 1, pp. 135-144, 2004.

[8] J. Gil, Y. Chan, C. Hwang, D. Park, J. Shon, and Y. Jeong, "Restoration Scheme of Mobility Databases by Mobility Learning and Prediction in PCS Networks," IEEE J. Selected Areas in Comm., vol. 19, no. 10, pp. 1962-1973, 2001.

[9] S. E. Fahlman and C. Lebiere, “The Cascade-Correlation Learning Architecture, in Neural Information Processing Systems”, Editors D. Touretzky, Morgan Kaufmann Publishers, Inc., Denver, Colorado, 1990, pp. 524-532.

[10] H. Hwang, M. Chang, and C. Tseng, "A Direction-Based Location Update Scheme with a Line-Paging Strategy for PCS Networks," IEEE Comm. Letters, vol. 4, no. 5, pp. 149-151, May 2000.

[11] A. Aljadhai and T. Znati, "Predictive Mobility Support for QoS Provisioning in Mobile Wireless Environments," IEEE J. Selected Areas in Comm., vol. 19, no. 10, pp. 1915-1930, 2001.

[12] C. Wu, H. Lin, and L. Lan, "A New Analytic Framework for Dynamic Mobility Management of PCS Networks," IEEE Trans. Mobile Computing, vol. 1, no. 3, pp. 208-220, 2002.

[13] S. Halpin and R. Burch, "Applicability of Neural Networks to Industrial and Commercial Power Systems: A Tutorial Overview," IEEE Trans. Industry Applications, vol. 33, no. 5, pp. 1355-1361, 1997.

[14] G. Pollini and I. Chih-Lin, "A Profile-Based Location Strategy and Its Performance," IEEE J. Selected Areas In Comm., Vol. 15, No. 8, pp. 1415-1424, 1997.

[15] Wenye Wang, Ian F. Akyildiz and Gordon L. Stiiber, “An Optimal Paging Scheme for Minimizing Signaling Costs under Delay Bounds”, IEEE Communications Letters, 2001.

Karnaugh Map Model for Mining Association Rules in Large Databases

Neelu Khare[1], Neeru Adlakha[2] and K. R. Pardasani[3]

[1] Department of Computer Applications
Maulana Azad National Institute of Technology
Bhopal (M.P.) India
neelukh_29@yahoo.com

[2] Department of Applied Mathematics
SVNIT, Surat,Gujrat
neeru.adlakha2@gmail.com

[3] Departmet of Mathematics
Maulana Azad National Institute of Technology
Bhopal (M.P.) India
kamalrajp@hotmail.com

Abstract: ***Mining frequent patterns is an important component of association rule mining. A number of attempts have been made by investigators in the past to develop models and algorithms for association rule mining but these algorithms suffer from major setback of number of database scans. The large number of database scans required in Apriori algorithm makes this mining process slow. Improvements have been made by investigators and they have succeeded in reducing the number of database scans to two. Here an attempt has been made to develop a model which requires less than two database scans. A Karnaugh Map model has been developed to compress the whole database in terms of frequency of item sets. Thus the Karnaugh Map matrices will have size very less than that of whole database for the mining process carried on the Karnaugh Map matrix. Thus the whole database will be scanned only once and the Karnaugh Map matrices will have size equivalent to very small fraction of whole database. Thus this approach brings efficiency in association rule mining.***

Keywords: Karnaugh Map model**,** frequent patterns, Association rules, Cartesian product, frequency matrix.

1. Introduction

Data mining and knowledge discovery (KDD) lies at the interface of statistics, database technology, machine learning, high performance computing etc [17]. It is concerned with the computational and data intensive process of deriving interesting and useful information or patterns from massive databases. Association rule mining is an important data mining technique to generate correlation and association rule. The problem of mining association rules could be decomposed into two sub problems, the mining of large itemsets (i.e. frequent itemsets) and the generation of association rules[1][3].Data mining is motivated by decision support problems faced by most business organizations and is described as an important area of research. The main challenges in database mining are developing fast and efficient algorithms,that can handle large volume of data because the most mining algorithms perform computation over the entire database and often the database is very large[4].

Many mining algorithms have been developed for discovering association rules [3, 4, 6]. One of the key features of all the previous algorithms that they require multiple passes over the database. For disk resident databases, this requires reading the database completely for each pass resulting in a large number of disk I/Os.

The issues are-

1. There is no enough time to perform a multiple scan whenever an update occurs.
2. A one scan of data and compact memory usage of the association rule mining technique are necessary.
3. A mining mechanism that adapts itself to available resources is needed [3] [5].

There has been a lot of research in developing efficient algorithms for mining frequent itemsets. Most of them enumerate all frequent itemsets[12,13]. There also exist methods which only generate frequent closed itemsets and maximal frequent itemsets[10]. While these methods generate a reduced number of itemsets, they still need to mine the entire database in order to generate the model of frequent itemsets, and therefore these methods are not efficient in mining evolving databases.[1][9][11]

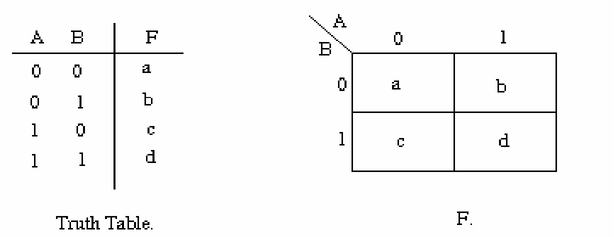
Aproiri based algorithms require multiple scans of the original database, which leads to high CPU and I/O costs. The Apriori Algorithm finds out the large itemsets iteratively. Apriori-like algorithms iteratively obtain candidate itemsets of size (k+l) from frequent itemsets of size k. Each iteration requires a scan of the original database. It is costly and inefficient to repeatedly scan the database and check a large set of candidates for their occurrence frequencies[21]. Additionally, when new data comes in, we have to run the entire algorithms again to update the rules. There are various methods to improve Apriori performance[3]. While these measures improve the situation, there are still many problems. Recently, an FP-tree based frequent patterns mining method developed by Han achieves high efficiency as compared with Apriori" and

Tree Projection algorithms. It avoids iterative candidate generations. But it suffer from following problems:
1. Creating conditional FP-Tree separately and recursively during the mining process.
2. Updating of FP-tree requires two scans of the new data and existing data[20].
3. Pointers are to be used in tree structure that will increase storage overheads.
4. Every time when an update occurs pointer adjustment is required .
5. When the number of items becomes large then generating tree patterns is very complex.[11][10][3]

In this paper an attempt has been made to develop a new approach for mining association rule in large databases. This approach compresses the database in the form of frequency, which will reduce the size of database and the mining is performed on the compressed data set. Karnaugh Map technique has been used to achieve this compression which stores all of the information in a highly compact form and updates easily. Then set theory is used to calculate support count for itemsets. Thus, mining requires only one scan of the database and updating Trans-Tree needs one scan of the new data without scanning the existing data. The frequent itemsets are extracted from Karnaugh Map.

1.1 Karnaugh Map

A Karnaugh map provides a pictorial method of grouping together expressions with common factors and therefore eliminating unwanted variables. The Karnaugh map can also be described as a special arrangement of a truth table. The diagram below illustrates the correspondence between the Karnaugh map and the truth table for the general case of a two variable problem. [7]



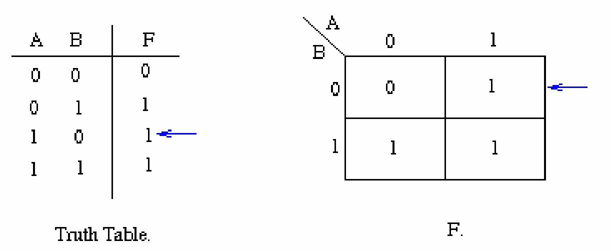
A	B	F
0	0	a
0	1	b
1	0	c
1	1	d

Truth Table.

B \ A	0	1
0	a	b
1	c	d

F.

The values inside the squares are copied from the output column of the truth table, therefore there is one square in the map for every row in the truth table. Around the edge of the Karnaugh map are the values of the two input variable. A is along the top and B is down the left hand side. The diagram below explains this:



A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table.

B \ A	0	1
0	0	1
1	1	1

F.

The values around the edge of the map can be thought of as coordinates. So as an example, the square on the top right hand corner of the map in the above diagram has coordinates A=1 and B=0. This square corresponds to the row in the truth table where A=1 and B=0 and F=1. Note that the value in the F column represents a particular function to which the Karnaugh map corresponds [7]. In a Karnaugh map with *n* variables, a Boolean term mentioning *k* of them will have a corresponding rectangle of area 2^{n-k}. Common sized maps are of 2 variables which is a 2x2 map; 3 variables which is a 2x4 map; and 4 variables which is a 4x4 map.

2. Model and Method

In view of the above mentioned issues a model have been developed using Karnaugh Map and set theory. Let I = {$i_{1,}$ i_2, i_3,…,i_n} be the set of all items, where each item is a binary variable. It can hold either **0** or **1** at a time. T = {$t_{1,}$ t_2, t_3,.….,t_n}be the set of all transactions. Each transaction t_i contains a subset of items chosen from **I**. In association a collection of 0 or more items is termed an itemset. If an itemset contains k items, it is called K-itemset. For instance{$i_{1,}$ i_2, i_3} is an example of 3-itemset. The null (or empty)set is an itemset that does not contain any item.It is assumed that the size of largest itemset is known and the Karnaugh map of that size will be created. All the items will be treated as binary variables and indicate the presence and absence of an item. Let the size of largest itemset is n, then n variable Karnaugh map will be designed on the basis of which a matrix X will be created.
Total numbers of possible itemsets with n item variables are 2^n -1. **Table1** shows the possible size and number of itemsets of that size for a given number of items(e.g, **n**).

Table1: Possible Size and Number of items

Number of items in the itemset	**Number of possible combinations**	**Possible number of itemsets**
1	${}^nC_1$	N
2	${}^nC_2$	N(n-1)/2!
3	${}^nC_3$	N(n-1)(n-2)/3!
:	:	:
:	:	:
:	:	:
N	${}^nC_n$	1

The set of all possible item sets, for a given number of items can be defined as power set of the set I, denoted by ρ**(I)**, having 2^n elements including empty set. Then check whether **n** is even or odd.
If n is even then a matrix of $\mathbf{2^{n/2}\ X\ 2^{n/2}}$ is created.
If n is odd then a matrix of $\mathbf{2^{(n-1)/2}\ X\ 2^{(n+1)/2}}$ is created.

Karnaugh map takes **n/2** or **(n-1)/2** items and their combinations along with rows and **n/2** or **(n+1)/2** items and their combinations along with columns ; we call these items row variables and column variables respectively . So there are $\mathbf{2^{n/2}}$ or $\mathbf{2^{(n-1)/2}}$ itemsets along rows and $\mathbf{2^{n/2}}$ **or** $\mathbf{2^{(n+1)/2}}$ itemsets along columns.The matrix can be implemented by two dimensional integer array, each element of that will store the frequency of an itemset. Initially all the array elements store zero. Data will be stored in the form of frequency in the matrix. In a particular itemset the presence of an item is indicated by the value 1 and absence of an item by 0.

For each itemset there will be a specified location in the matrix according to the Karnaugh map logic that is on the basis of the items present in the itemset. The presence of that item in a set is marked by the value 1 (otherwise 0) held by it. Logically each item set will be considered as a string of binary digits , that string will decide position of that itemset in the Karnaugh map. Whenever a transaction occurs its respective array element's value will be incremented by 1. At each transaction array elements are checked and updated. So at any instant the updated information of itemsets are available. Where there is no need to scan the database many times. Only one scan is sufficient to store and update the database. This algorithm will also use a priori knowledge of frequent item sets properties.

A matrix contains rows and columns; we assume that matrix has a set of rows R and a set of columns C. The set
$R = \{r_{1,}\ r_{2,}\ r_{3..........}r_{x\}}$ consists of rows in matrix where r_i represents the i^{th} row in R($x = 2^{n/2}$ or $2^{(n-1)/2}$).

$r_{i=(}\ a_{i1} \wedge a_{i2} \wedge a_{i3}............ \wedge\ a_{ij)}$ where $r_i \in R$

Similarly set $C = \{c_{1,}\ c_{2,}\ c_{3..........}c_{y\}}$ consists of columns in matrix where c_j represents the j^{th} column in C ($y = 2^{n/2}$ or $2^{(n+1)/2}$).

$c_{j\,=\,(}\ a_{ij} \wedge a_{2j} \wedge a_{3j}............ \wedge\ a_{ij)}$ where $c_j \in C$

Each row column combination specifies the location of a cell in matrix (e.g; a_{21} cell belongs to 2nd row 1st column) which stores the frequency of an allocated itemset. So item variables can be divided into two categories row variables set RVar and Column variables set CVar. The possible combinations of items formed by Rvar and Cvar, denoted by the elements of the sets Rset and Cset respectively and these Rset and Cset will be the power set of Rvar and Cvar sets.

Rvar = { i: is an k^{th} element of I (k is within 1…n/2 or (n-1)/2)}

Cvar = { i: is an p^{th} element of I (p is within n/2 or (n-1)/2)…….(n+1)/2}

Possible combinations of Rvar and Cvar are in the set

Rset = ρ (Rvar)

Cset = ρ (Cvar)

Where Rset $\subseteq \rho$(I)

Cset $\subseteq \rho$(I) and Rset $\cap$ Cset = ϕ

Rest of the itemset combinations (of ρ**(I))** will be found in the Cartesian product set of the sets Rset and Cset (Rset×Cset). From the above assumption we can see that all the elements of Rset lies in the rows of set R, so there will be one to one correspondence between the elements of R and Rset, similarly between C and Cset also. According to apriori principal 'Every subset of a frequent itemset has to be frequent.', on the basis of this we first count the support for single items (they found in Rset and Cset) and compare with the threshold, then the items those are frequent, we consider their supersets only in next step. The support count for a single item of Rset / Cset, will be the sum of all cell elements of its respective row/column and its superset's row/column. It can be calculated as union of their container rows and columns.

$\mathrm{Supp}(I_k) = \{\sum a_{ij} : \forall\ a_{ij} \in (r_i \cup r_{i+1} \cup ....r_x)\}/N$ …(1)

Where $I_k \in$ Rset

$\mathrm{Supp}(I_k) = \{\sum a_{ij} : \forall\ a_{ij} \in (c_i \cup c_{i+1} \cup ....c_y)\}/N$ …(2)

Where $I_k \in$ Cset

N is total number of transactions in the table.

By the above formulas we can calculate the support of all single itemsets. Then generate frequent single item sets, only frequent items will be considered to generate 2-itemset in next step. Here itemsets will be the elements of Rset, Cset and (Rset×Cset). The itemsets which belongs to Rset and Cset , their support can be calculated by the sum of the cell values of the intersection of their container rows and columns respectively.The itemsets which are elements of Rset and Cset their support can be calculated by;

$\mathrm{Supp}(I_k\, I_{k+1}) = \{\sum a_{ij}: \forall\, a_{ij} \in (r_i \cap r_{i+1} \cap ......)\}/N$ …(3)

Where $(I_k\, I_{k+1}) \in$ Rset

$\mathrm{Supp}(I_k\, I_{k+1}) = \{\sum a_{ij}: \forall\, a_{ij} \in (c_i \cap c_{i+1} \cap ......)\}/N$ …(4)

Where $(I_k\, I_{k+1}) \in$ Cset

Now rest of the itemsets belong to the set (Rset×Cset) and their count will be calculated as the sum of the matrix elements that belongs to the Cartesian product set of the involved item/itemset's row and column. The itemsets those are elements of (Rset×Cset) their support can be calculated by ;

$\mathrm{Supp}(I_k I_l) = |\{\sum a_{ij}: \forall\, a_{ij} \in$ (container rows of $I_k \times$ container column of $I_l......)\}|/N$ …(5)

We take an example where n = 4. The items are
$I = \{i_{1,}\ i_2,\ i_3,\ i_4\}$. T is the set of transactions T={T1,T2……T20} (Table5 is the transaction table) .

Following table shows the number of possible itemsets in the order of items. (Bold letters shows that item is present and assigned the value 1 Letters with 'shows complement of the variable or shows absence of the variable).

Table2: Possible itemsets

Number of items in the itemset	**Number of possible combinations**	**Possible number of itemsets**
1	$^4C_1 = 4$	**I1** I2' I3' I4' I1' **I2** I3' I4' I1' I2' **I3** I4' I1' I2' I3' **I4**
2	$^4C_2 = 6$	**I1 I2** I3' I4' I1' I2' **I3 I4** I1' **I2 I3** I4' **I1** I2' I3' **I4** **I1** I2' **I3** I4' I1' **I2** I3' **I4**
3	$^4C_3 = 4$	**I1 I2 I3** I4' I1' **I2 I3 I4** **I1** I2' **I3 I4** **I1 I2** I3' **I4**
4	$^4C_4$	**I1 I2 I3 I4**

So a 4 variable Karnaugh map will be designed and implemented by a matrix of $\mathbf{2^{n/2}\ X\ 2^{n/2}}$ i.e., matrix of $\mathbf{2^2} \times \mathbf{2^2} = 4 \times 4$ matrix will be created. (Table3)

X[4][4] i.e., X[0…3][0…3]. The numbers in the matrix shows the numerical assignment of the cells.

Table3: First n/2 Variables are taken along Rows and Next n/2 Variables are Along with Columns

I3I4 I1I2	c_1 **00**	c_2 **01**	c_3 **10**	c_4 **11**
r_1 00	**0**	**1**	**2**	**3**
r_2 01	**4**	**5**	**6**	**7**
r_3 10	**8**	**9**	**10**	**11**
r_4 11	**12**	**13**	**14**	**15**

From the table we can see that first n/2 variables are taken along rows and next n/2 variables are along with columns.

Rvar = { I1, I2} and Cvar = { I3, I4}

Rset= ρ (Rvar)= { { },{I2},{I1},{I1 I2} }

Cset= ρ (Cvar)= { { },{I4},{I3},{I3 I4} }

(Rset×Cset)={{I2I4},{I2I3},{I2I3I4},{I1I3}, {I1I4}, {I1I3I4}, {I1I2I4}, {I1I2I3}, {I1I2I3I4}}

So Rset, Cset and (Rset×Cset) are subset of ρ (I) and Rset $\cup$ Cset $\cup$ (Rset×Cset) = ρ (I)

And Rset $\cap$ Cset $\cap$ (Rset×Cset) = ϕ. Matrix contains set of rows R = {$r_{1,}$ $r_{2,}$ r_3 ,r_4} and set of columns C = {$c_{1,}$ $c_{2,}$ $c_{3,}c_4$}.

The Support count of Single item sets are calculated by :

Supp(I1) = $|[\sum a_{ij} : \forall \; a_{ij} \in (r_3 \cup r_4)] \; |/N$ *(for i=3,4 j=1..4) ....(5a)*

Supp(I1) = $[(a_{31} + a_{32} + a_{33} + a_{34}) + (a_{41} + a_{42} + a_{43} + a_{44} /N$*................(5b)*

Supp(I2) = $|[\sum a_{ij} : \forall \; a_{ij} \in (r_2 \cup r_4)] \; |/N$ *(for i= 2,4 j=1..4).............(6a)*

Supp(I2) = $[(a_{21} + a_{22} + a_{23} + a_{24}) + (a_{41} + a_{42} + a_{43} + a_{44})]/N$*..(6b)*

Supp(I3) = $|[\sum a_{ij} : \forall \; a_{ij} \in (c_3 \cup c_4)] \; |/N$ *(for i=1..4 j=3,4) ....(7a)*

Supp(I3)=$[(a_{13} + a_{23} + a_{33} + a_{43})+(a_{14} + a_{24} + a_{34} + a_{44})]/N$*.............(7b)*

Supp(I4) = $|[\sum a_{ij} : \forall \; a_{ij} \in (c_2 \cup c_4)] \; |/N$ *(for i=1..4 j=2,4) ....(8a)*

Supp(I3)=$[(a_{12} + a_{22} + a_{32} + a_{42})+(a_{14} + a_{24} + a_{34} + a_{44})]/N$*..............(8b)*

After counting the support of single items we generate frequent single items and only their 2-itemsets will be considered in next step. The 2-itemsets which belongs to the Rset and Cset their count will be the sum of the cells contained in the intersection of their respective rows or columns, referred from the above equations.

{I1I2}$\in$ Rset

Supp(I_1I_2) ={$\sum a_{ij}$: $\forall$ $a_{ij} \in$ (container rows of I1 $\cap$ container rows of I2)/N

$$\text{Supp}(I_1I_2) = \{ \sum a_{ij}: \forall a_{ij} \in ((r_3 \cup r_4) \cap (r_2 \cup r_4))\}/N$$
$$= \{ \sum a_{ij}: \forall a_{ij} \in (r_4)\}/N$$
$$= (a_{41} + a_{42} + a_{43} + a_{44})/N$$

{I3I4}$\in$ Cset

Supp(I_3I_4)={$\sum a_{ij}$: $\forall$ $a_{ij} \in$ (container columns of I3 $\cap$ container columns of I4)/N

$$\text{Supp}(I_3I_4)=\{ \sum a_{ij}: \forall a_{ij} \in ((c_3 \cup c_4) \cap \in (c_2 \cup c_4))\}/N$$
$$= \{ \sum a_{ij}: \forall a_{ij} \in (c_4)\}/N$$
$$= (a_{14} + a_{24} + a_{34} + a_{44})]/N$$

The support count of the itemsets which are the elements of (Rset×Cset) can be calculate by

Supp(I_kI_l) = |{$\sum a_{ij}$: $\forall$ $a_{ij} \in$ (container row(s) of $\mathbf{I}_k \times$ container column(s) of I_l}|/N …(5)

$$\text{Supp}(I1I3) = \{ \sum a_{ij}: \forall a_{ij} \in (r_3 , r_4 \mathbf{)} \times (c_3 \; c_4)\}/N$$
$$= (a_{33} + a_{34} + a_{43} + a_{44})/N$$

Similarly support of all the itemsets can be calculated. The following table shows the decimal number for an itemset according to which the cell will be allocated for it.

Table4: The Decimal Number for an itemset According to which the Cell will be allocated for it

Itemset	**Position Value in Karnaugh map Decimal (Binary)**	**Assigned array element**
Single		
I1 (or **I1** I2' I3' I4')	8 (1 0 0 0)	X[2][0]
I2 (or I1' **I2** I3' I4')	4 (0 1 0 0)	X[1][0]
I3 (or I1' I2' **I3** I4')	2 (0 0 1 0)	X[0][2]
I4 (or I1' I2' I3' **I4)**	1 (0 0 0 1)	X[0][1]
Double		
I1I2(or **I1 I2** I3' I4')	12 (1 1 0 0)	X[3][0]
I3I4(or I1' I2' **I3 I4)**	3 (0 0 1 1)	X[0][3]
I2I3(or I1' **I2 I3** I4')	6(0 1 1 0)	X[1][2]
I1I4(or **I1** I2' I3' **I4)**	9(1 0 01)	X[2][1]
I1I3(or **I1** I2' **I3** I4')	10(1 010)	X[2][2]
I2I4(or I1' **I2** I3' **I4)**	5 (0 1 0 1)	X[1][1]
Triple		
I1I2I3(or**I1 I2 I3** I4')	14 (1110)	X[3][2]
I2I3I4**(or** I1' **I2 I3 I4)**	7 (0 1 11)	X[1][3]
I1I3I4**(or I1** I2' **I3 I4)**	11(1 0 1 1)	X[2][3]
I1I2I4**(or I1 I2** I3' **I4)**	13(1 1 0 1)	X[3][1]
Quad **(I1 I2 I3 I4)**	15(1 1 1 1)	X[3][3]

3. Result

The following data set has been used to implement the algorithm.

Table5: Data Set

S.No.	TID	Itemset
1	T1	I1
2	T2	I1,I3
3	T3	I1,I2,I3
4	T4	I1,12,I3
5	T5	I1,I2,I3,I4
6	T6	I3
7	T7	I1
8	T8	I1,I3
9	T9	I1,I3,I2
10	T10	I1
11	T11	I1
12	T12	I2,I4
13	T13	I3,I4
14	T14	I1,I2,I3
15	T15	I1,I2
16	T16	I1,I3
17	T17	I2,I3,I4
18	T18	I1,I2,I4
19	T19	I1,I2,I3,I4
20	T20	I2,I3,I4

On the basis of transaction Table5 Karnaugh map matrix is obtained (Table 6). The numbers in the cells show the frequency of their allocated item set.

Table6: Karnaugh Map Matrix for Dataset

I3I4 I1I2	C1 00	I4 C2 01	I3 C3 10	I3I4 C4 11
R1 00	0	0	1	1
R2 01	0	1	0	2
R3 10	4	0	3	0
R4 11	1	1	4	2

Step1- First we calculate the support count of Single items. They all are the elements of either Rset or Cset . Let the *min_sup*=0.35

From (5b)

Supp(I1) = [(4 +0 +3+0)+(1+1+4 +2)] /20
= 15/20=0.75

From (6b)

Supp(I2) = [(0+1+0+2) +(1+1+4 +2)] /20
= 11/20=0.55

I1,I2 $\in$ Rset

From (7b)

Supp(I3) = [(1+0+3+4)+(1 +2+0+2)]/20
= 13/20=0.65

From (8b)

Supp(I4) = [(0+1+0+1)+(1 +2+0+2)]/20
= 7/20=0.35

I3,I4 $\in$ Cset

Step2- Now compare the support count of single itemsets with *min_sup* to generate frequent single item.

Step3- all the single items have supp>*min_sup*. All 4 items are frequent.

Step4- calculate the support count for all possible 2-itemsets.

Supp(I1I2) = (a_{41} + a_{42} + a_{43} + a_{44})/N
= (1+1+4+2)/20
= 8/20 = 0.40

Supp(I3I4) = (a_{14} + a_{24} + a_{34} + a_{44})]/N
= (1+2+0+2)/20
= 5/20 = 0.25

Supp(I2I4) = $\{ \sum a_{ij}: \forall a_{ij} \in (r_{2\,,}\, r_4) \times (c_2 , c_4)\}$ /N
[(frrom (6a) and (8a)]
= (a_{22}+ a_{24}+ a_{42}+ a_{44})]/N
= (1+2+1+2)/20
= 6/20 = 0.30

Supp(I2I3) = $\{ \sum a_{ij}: \forall a_{ij} \in (r_{2\,,}\, r_4) \times (c_3 , c_4)\}$ /N
[(frrom (6a) and (7a)]
= (a_{23}+ a_{24}+ a_{43}+ a_{44})]/N
= (0+2+4+2)/20
= 8/20 = 0.40

Supp(I1I3) = $\{ \sum a_{ij}: \forall a_{ij} \in (r_{3\,,}\, r_4) \times (c_3 , c_4)\}$ /N
[(frrom (5a) and (7a)]
= (a_{33}+ a_{34}+ a_{43}+ a_{44})]/N
= (3+0+4+2)/20
= 9/20 = 0.45

Supp(I1I4) = $\{ \sum a_{ij}: \forall a_{ij} \in (r_{3\,,}\, r_4) \times (c_2 , c_4)\}$ /N
[(frrom (6a) and (8a)]
= (a_{32}+ a_{34}+ a_{42}+ a_{44})]/N
= (0+0+1+2)/20
= 3/20 = 0.15

Step5- Now we compare the support counts with min_supp to generate frequent 2-itemsets.(I1I2) (I1I3) (I2I3) (I2I4)are frequent itemsets and (I3I4) (I1I4) are not frequent. So in next step we will consider only the supersets of (I1I2) (I1I3) (I2I3) (I2I4) itemsets.

Step6- Sets (I1I2I3) (I1I2I4) (I2I3I4) are there, their support counts are :

Supp(I1I2I3) = $\{ \sum a_{ij}: \forall a_{ij} \in (r_4) \times (c_3 , c_4)\}$ /N
= (a_{43}+ a_{44})]/N
= (4+2)/20 = 0.30

Supp(I1I2I4) = $\{ \sum a_{ij}: \forall a_{ij} \in (r_4) \times (c_2 , c_4)\}$ /N
= (a_{42}+ a_{44})]/N
= (1+2)/20 = 0.15

Supp(I2I3I4) = $\{ \sum a_{ij}: \forall a_{ij} \in (r_{2,}\, r_4) \times (c_4)\}$ /N
= (a_{24}+ a_{44})]/N
= (2+2)/20 = 0.20

Step7- Now we compare the support counts with min_supp to generate frequent 3-itemsets. As none of the 3-itemset has supp>*min_supp*. So the frequent itemsets are (I1I2) (I1I3) (I2I3) (I2I4).

The conclusion should indicate the significant contribution of the manuscript with its limitations, advantages and applications.

4. Discussion

The algorithm developed here is quite efficient. It requires only one database scan and thus the I/O overheads have been reduced. It creates compressed data set which is a small fraction of original database and can easily fit into main memory. Thus processing can be done efficiently. Most of the time CPU will be busy computing the frequent itemsets from the compressed dataset in the form of frequency. Although it increases the processing overheads for CPU, but the speed of CPU is much higher(100 times) than the I/O speed of mining frequent itemsets very significantly. This algorithm can further be extended to other mining tasks, like incremental mining, continuous mining etc.

References

[1] Jurgen M. Jams Fakultat fur Wirtschafts- irnd, "An Enhanced Apriori Algorithm for Mining Multidimensional Association Rules", 25th Int. Conf. Information Technology interfaces ITI Cavtat, Croatia, 2003.

[2] Rolly Intan ,"A Proposal Of Fuzzy Multidimensional Association Rules", , Jurnal INFORMATIKA VOL 7,pp. 85-90, November 2006.

[3] Ravindra Patel, D. K. Swami and K. R. Pardasani ,"Lattice Based Algorithm for Incremental Mining of Association Rules", International Journal of Theoretical and Applied Computer Sciences, Volume 1 Number 1 Journal Computer Science, USA , pp. 119–128, 2006.

[4]Agrawal R., Imielinski T., and Swami A. "Mining Association rules between sets of items in large databases". In Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, Washington, USA , 1993.

[5] Nan Jiang and Le Gruenwalds," Research Issues in Data Stream Association Rule Mining" , SIGMOD Record, Vol. 35, No. 1, Mar. 2006.

[6] Agrawal R. and Srikant R. "Fast algorithms for rules Mining Association in large databases". In Proc. Of the Twentieth International Conference on Very Large Databases, pp. 487-499, 1994.

[7] "wilkipedia," [Online]. Available: www.wilkipedia.com [Accessed: Oct. 15, 2009].

[8] Cheung D., Lee S., and Kao B. "A general incremental technique for maintaining discovered association rules". In Proc. of the 5thIntl. Conf. on Database Systems for Advanced Applications, pp. 1–4 , 1997.

[9] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li.New "Algorithms for Fast Discovery of Association Rules". In Proc. of the 3rd Int'l Conference on Knowledge Discovery and Data Mining, Newport Beach, Cal-ifornia, Aug. 1997.

[10] Lee S. and Cheung D." Maintenance of discovered association rules When to update?" In Research Issues on Data Mining and Knowledge Discovery, 1997.

[11] Thomas S., Bodagala S., Alsabti K., and Ranka S. "An efficient algorithm for the incremental updating of association rules". In Proc. of the 3rd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, 1997.

[13] Veloso A., Meira Jr W., de Carvalho M. B., Possas B., Parthasarathy S., and Zaki M. "Mining frequent itemsets in evolving databases". In Proc. of the 2nd SIAM Int'l Conf. on Data Mining , Arlington, USA, 2002.

[12] F. Berzal, J.C. Cubero, N. Marin, J.M. Serrano, " An efficient method for association rule mining in relational databases", Elserier Data & Knowledge, Engineering, pp. 47–64, 2001.

[13] S. Brin, R. Motwani, C. Silverstein, "Beyond market baskets: generalizing association rules to correlations", ACM SIGMOD Conference on Management of Data, Tuscon, AZ, pp. 265–276, May 1997.

[14] D.W. Cheung, J. Han, V.T. Ng, A.W. Fu, Y. Fu, "A fast distributed algorithm for mining association rules", Proc. of Int'l Conf. on PDIS'96, Miami Beach, FL, USA, Dec. 1996.

[15] Zhang Hong, Zhang Bo , Kong Ling-Dong ,Cai Zheng-Xing, "Generalized Association Rule Mining Algorithms based on Data Cube," IEEE DOI 10.1109/SNPD, 2007.

[16] Jun Gao ,"Realization of a New Association Rule Mining Algorithm" IEEE DOI, 2007.

[17] Ashok Savasere,"An Efficient Algorithm for Mining Association Rules in Large Databases" GA 30332, 1995.

[18] Han, J., Pei, J., Yin, Y. "Mining Frequent Patterns without Candidate Generation",SIGMOD Conference, pp. 1-12, 2000.

[19] Han, J., Pei, J., Yin, Y., Mao, R. "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach". Data Mining and Knowledge Discovery, pp. 53–87, 2004.

[20] J. Han, M. Kamber, Data Mining: Concepts and Techniques, The Morgan Kaufmann Series, 2001.

[21] Klemetinen, L., Mannila, H., Ronkainen, P., "Finding interesting rules from large sets of discovered association rules". Third International Conference on Information and Knowledge Management Gaithersburg, USA, pp.401-407, 1994.

New Dynamic Approach for LZW Data Compression

Balwant Singh Raghuwanshi [1], Sonali Jain [2], Dushyant Chawda [3] and Bhupendra varma [4]
[1]PG Research Scholar in Department of computer science and Engineering T. I. T., Bhopal, India
balwant8336@gmail.com
[2]Assistant Professor in Department of E & TC, BIRT, Bhopal, India
sonali010@rediffmail.com
[3]Assistant Professor in Department of computer science and Engineering TIT, Bhopal, India
[4]Professor Computer science and Engineering TIT, Bhopal, India

Abstract: ***There are large numbers of compression techniques available which are capable of application specific large compression ratios. Here we limit our scope to the LZW algorithm, which can be considered as the industry standard for the loss less data compression. This paper presents a modified approach for LZW algorithm, which will incorporate. The dynamic restructuring of the number of bits and Feedback mechanism. The level of compression depending on the above factors which are crucial for working of LZW. LZW compression excels when confronted with data streams that have any type of repeated strings. Because of this, it does extremely well when compressing English text. Compression levels of at least 50% or better than 50% mainly achieved. Likewise, compressing saved screens and displays will generally show very good results.***

Keywords: LZW**,** Feedback mechanism, Dynamic Restructure of Bits, Compression Level**.**

1. Introduction

Data compression [5, 8] is the process of converting an input data stream (the source stream or the original raw data) into another data stream (the output, or the compressed stream) that has a smaller size [10, 11] A stream is either a file or a buffer in memory. A simple characterization of data compression is that it involves transforming a string of characters in some representation (such as ASCII) into a new string (of bits, for example) which contains the same information but whose length is as small as possible. Data compression has important application in the areas of data transmission and data storage. Many data processing applications require storage of large volumes of data, and the number of such applications is constantly increasing as the use of computers extends to new disciplines. At the same time, the proliferation of computer communication networks is resulting in massive transfer of data over communication links. Compressing data to be stored or transmitted reduces storage and/or communication costs. When the amount of data to be transmitted is reduced, the effect is that of increasing the capacity of the communication channel. Similarly, compressing a file to half of its original size is equivalent to doubling the capacity of the storage medium. It may then become feasible to store the data at a higher, thus faster level of the storage hierarchy and reduce the load on the input/output channels of the computer system. In order to discuss the relative merits of data compression techniques, a framework for comparison must be established. There are two dimensions along which each of the schemes discussed here may be measured, algorithm complexity and amount of compression. When data compression is used in a data transmission application, the goal is speed. Speed of transmission depends upon the number of bits sent, the time required for the encoder to generate the coded message, and the time required for the decoder to recover the original ensemble. In a data storage application, although the degree of compression is the primary concern, it is nonetheless necessary that the algorithm be efficient in order for the scheme to be practical.

2. LZW Coding Technique

The original Lempel Ziv approach to data compression was first published in in1977. Terry Welch's refinements to the algorithm were published in 1984. The algorithm is surprisingly simple. In a nutshell, LZW compression replaces strings of characters with single codes [4, 13, 14]. It does not do any analysis of the incoming text. Instead, it just adds every new string of characters it sees to a table of strings. Compression occurs when a single code is output instead of a string of characters. The code that the LZW algorithm outputs can be of any arbitrary length, but it must have more bits in it than a single character. The first 256 codes (when using eight bit characters) are by default assigned to the standard character set. The remaining codes are assigned to strings as the algorithm proceeds. The sample program runs as shown with 12 bit codes. This means codes 0-255 refer to individual bytes, while codes 256-4095 refers to substrings.

2.1 Compression

The LZW compression algorithm in its simplest form is shown below. A quick examination of the algorithm shows that LZW is always trying to output codes for strings that are already known. And each time a new code is output, a new string is added to the string table [4].

ENCODING:

```
 STRING = get input character
   WHILE there are still input characters DO
       CHARACTER = get input character
       IF STRING+CHARACTER is in the string table then
           STRING = STRING + character
       ELSE
           Output the code for STRING
           Add STRING+CHARACTER to the string table
           STRING = CHARACTER
       END of IF
   END of WHILE
   Output the code for STRING
```

2.2 Decompression

The companion algorithm for compression is the decompression algorithm. It needs to be able to take the stream of codes output from the compression algorithm, and use them to exactly recreate the input stream. One reason for the efficiency of the LZW algorithm is that it does not need to pass the string table to the decompression code. The table can be built exactly as it was during compression, using the input stream as data. This is possible because the compression algorithm always outputs the STRING and CHARACTER components of a code before it uses it in the output stream. This means that the compressed data is not burdened with carrying a large string translation table.

There is a single exception case in the LZW compression algorithm that causes some trouble to the decompression side. If there is a string consisting of a (STRING, CHARACTER) pair already defined in the table, and the input stream then sees a sequence of STRING, CHARACTER, STRING, CHARACTER, STRING, the compression algorithm will output a code before the decompress or gets a chance to define it [4].

DECODING:

```
Read OLD_CODE
Output OLD_CODE
CHARACTER = OLD_CODE
WHILE there are still input characters DO
   Read NEW_CODE
   IF NEW_CODE is not in the translation table THEN
       STRING = get translation of OLD_CODE
       STRING = STRING+CHARACTER
   ELSE
       STRING = get translation of NEW_CODE
   END of IF
   Output STRING
   CHARACTER = first character in STRING
   Add OLD_CODE + CHARACTER to the translation
 table
   OLD_CODE = NEW_CODE
END of WHILE
```

3. Factors That Affect the Formance of LZW

Based on the study of various algorithms following factors are identified which are crucial for working of LZW
[11, 14, 15]

* Number of bits to represent a dictionary
  Code word and the maximum size
  of the dictionary.
* Dynamically restructuring of the number of
  Bits to represent a code word depending
  Upon its magnitude.
* Static or dynamic nature of the Dictionary.

Now we will discuss each one of the above in some detail.

3.1 Maximum Size of the Dictionary

The maximum size of the dictionary plays an important role in the sense that the larger the size the greater number of bits will be required to represent a single character or a code word. This would mean if the size of the file, which needs to be compressed were small. Thus the large size of the dictionary will not be optimally used and will end in the increased size of the compressed file.

Thus the dictionary size should be based on the size of the file that needs to be compressed.

3.2 Dynamic Restructuring of the Number of bits

Dynamic restructuring of the number of bits to represent the code word would enable us to save those extra bits that need not be used by the code word which lies in the range that can be represented using the lesser number of bits .The standard LZW uses a fixed number of bits to represent each code thus if the size of the dictionary is 4096, then it will use 12-bits ($\log_2 4096$) for each code even if the code is 268(say) which can be represented by 9-bits.Thus dynamic restructuring of the number of bits helps us to save this loss.

3.3 Dynamic Nature of Dictionary

Dynamic nature of dictionary can be employed, which will be based on the feedback from the compression algorithm. The feedback parameter will be the current compression ratio relative to the threshold. In the dynamic dictionary the compression ratio will be the continuously matched against a threshold and if the ratio goes beyond what is specified in the threshold, the dictionary is flushed at that moment itself and a new one is created from the remaining inputs.

Thus if dynamic nature is employed the additional factors that need to be analyzed for their effect on the LZW are:

*** The threshold value**
***The compression check period.**

So far as threshold value is concerned it is not necessary that a higher threshold will result in higher compression always. It all depends on the contents of the file that is to be compressed. At times it might happen that if a very large threshold value is specified, the dictionary would be frequently flushed (since the compression ratio doesn’t match) and this frequent flushing may result in a compression ratio which would be smaller than the optimum which could have been achieved using a lower threshold value. Like the threshold value the compression check period also plays an important role. The check period implies the duration of characters after which the current compression ratio is matched against the threshold value. The compression check period is critical in the sense that if the compression check period is too large than by the time it would realize that the compression ratio has gone down, it would have deviated a lot from the threshold and then there is no way to revert back. Similarly if the compression checks period is too small then the moment it realizes that the compression ratio is going down there may such sequences immediately following, which could lead to the increase in the compression ratio.

4. The Proposed Compression Method

Based on the above-discussed parameters we propose a modified approach, which will incorporate the following:
*The dynamic restructuring of the number of bits (based on the absolute value of the sequence number of the current output).
*Feedback mechanism.

4.1 Incorporating the Dynamic Restructuring of the Number of bits:

If normal restructuring of the number of bits is applied on the basis of the absolute value of the index of the current entry in the dictionary then a significant number of bits is saved. For example suppose we completely fill a 4096 size dictionary this would mean that we use 8-bits up to 255, 9bits up to 511, 10 bits up to 1023, 11 bits up to 2047 and 12 bits only for the range 2048-4095. Thus in effect the total number of bits saved as compared the approach without the dynamic restructuring of the number of bits will be. 12*4096-(8*256+9*256+10*512+11*1024+12*2048)
which is equal to 3840 bits thus quite an achievement.

Generally the benefit will be more as the total input to be scanned would not be such that its contents will be such that they will only saturate the dictionary and no input will be left which could be scanned. For several hundred Kbytes of information we can even increase the dictionary size to 15 bits. To let the decompression program know when the bit size of the output code is going to change, a special BUMP_CODE is used. This code tells the decompression program to increase the bit size immediately. Another variant on LZW compression method is to build a phrase by concatenating the current phrase and the next character of data. This causes a quicker buildup of longer strings at the cost of a more complex data dictionary an alternative method would be to keep track of how frequently strings are used, and to periodically flush values that are rarely used. An adaptive technique like this may be too difficult to implement in a reasonably sized program. One final technique for compressing the data is to take the LZW codes and run them through an adaptive Huffman coding filter. This will generally exploit a few more percentage points of compression, but at the cost of considerable more complexity in the code, as well as quite a bit more run time.

4.2 Using Feedback

The standard LZW uses a single dictionary, i.e. the dictionary is only created once, if it gets filled then no more patterns can be formed and hence the only replaceable patterns are the ones which exist in the dictionary. What we mean to say that once the dictionary is full the input read is compared to the patterns within the dictionary if matched they are replaced by the appropriate code word otherwise the characters are outputted as such. This approach followed by the standard LZW at times leads to reduction in the compression ratio. To overcome the above stated drawback we suggest dynamic feedback to the algorithm. The feedback is in the sense that we keep monitoring the compression ratio at suitable intervals, after each such interval the compression ratio is compared against a threshold value, if the compression ratio is below the threshold the dictionary is flushed out (FLUSH_CODE is used here) and a new one is created which helps us to improve the compression ratio.

5. Application Constraints

The application constraints, which are applicable to the LZW, are also applicable to our modified version. It is particularly suitable for text files and the performance cannot be guaranteed for the image files. The compression ratio becomes better with the increase in the size of the source file.

6. Results

It is somewhat difficult to characterize the results of any data compression technique. The level of compression achieved varies quite a bit depending on several factors. LZW compression excels when confronted with data streams that have any type of repeated strings. Because of this, it does extremely well when compressing English text. Compression levels at least 50% or better should be expected. Likewise, compressing saved screens and displays will generally show very good results. We applied this approach on various types file like Text, Image and Sound and results are very good with respect to compression ratio.
We present our experimental result in the form of table.

Table 1: The compression effect on the size of Text files

Type of file	**Starting size in (kb)**	**Compressed size in (kb)**	**Compression Ratio (%)**
Sample1.doc	120	67.4	44
Sample2. doc	121	64	47
Sample3. doc	883	453	49
Sample4. doc	46.5	20.8	56
Sample5. doc	103	43.9	58
Sample6. doc	37.5	9.56	75

Compression Graph of Text File:

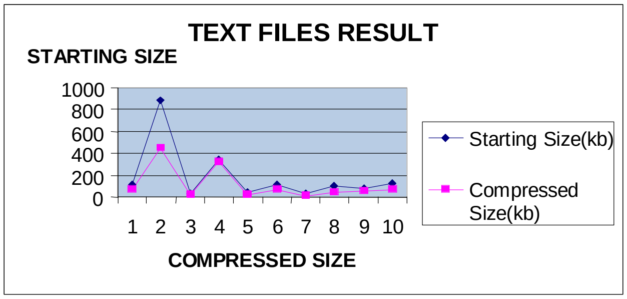


Figure1. Effect of the size on compression for Text file

Table 2: The Compression effect on the size of Image Files:

Type of file	Starting size in (kb)	Compressed size in (kb)	Compression Ratio (%)
Image1.cpt	170	68.4	59.76
Image2.cpt	74	21.6	70.86
Image3.cpt	153	39.7	70.05
Image4.cpt	764	68.4	83.9

Compression Graph of Image File:

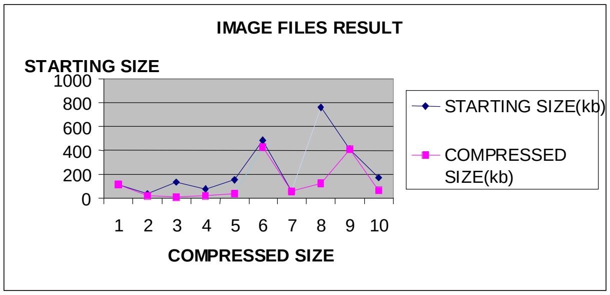


Figure2. Effect of the size on compression for Image file
Compression Graph of Sound File

Table 3: The Compression effect on the size of sound Files

Type of file	Starting size in (kb)	Compressed size in (kb)	Compression Ratio (%)
Sound1.wav	147	75	48.9
Sound2.wav	4.39	1.9	56.7
Sound3.wav	1260	468	62.85
Sound4.wav	1260	368	70.7
Sound5.wav	1260	111	91.19

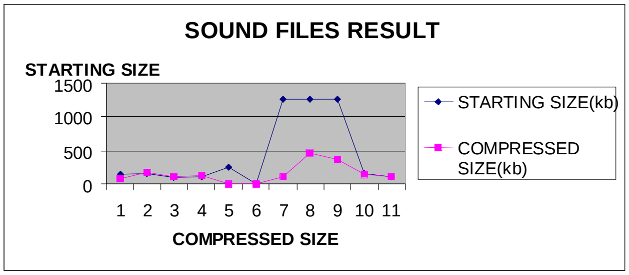


Figure3. Effect of the size on compression for Sound file

7. Conclusions

The optimization have been implemented and tested on variety of formats. The results, although they are for a limited number of formats, prove our optimization worthy. It is absolutely clear that for the same set of data files our approach leads to a more compressed file even if the dictionary size is made constant. Moreover it was also observed that for the same file varying the dictionary size could vary the compression. Compression threshold and compression check period. Thus this proves our prediction of the factor, which affects the compression ratio to be correct.

References

[1] Muhammad Younus Javed and Mr. Abid Nadeem, “Data Compression through Adaptive Huffman Coding Scheme”, IEEE, 2000.

[2] D.A. Huffman, A method for the constraction of Minimum Redundancy Codes, Proceedings of the IRE, pp. 1098-1101, 1952.

[3] Vitter, S. V., “Design and analysis of dynamic Huffman codes” ,Journal of the Assocition for Computing Machinery ,Vol. 34, No. 4,pp. 825-845, 1987.

[4] J. Ziv and A. Lempel, “A Universal Algorithm for Sequential Data Compression”, IEEE Transactions on Information Theory, May 1977.

[5] Khalid Sayhood, “Introduction to Data Compression”, Morgan Kaufmann, 2000. TK 5102.92.S39, March 2000.

[6] Dongyang Long, Weijia Jia,”Optimal Maximal and Maximal Prefix codes Equivalent to Huffman Codes”, Vol. III, pp.2525-2528, IEEE –2002.

[7] E. Bocharova, R. Johannesson, andB. D. Kudryashov, “Low state complexity block codes via convolutional codes,” *IEEE Trans. Inf.Theory*, Vol. 50, No. 9, pp. 2022–2030, Sep. 2004

[8] Data Compression Conference (DCC'00), Snowbird, Utah, March 2000.

[9] Lonardi, S and Szpankowski,W “Join Source –Channel LZ’77coding”, In IEEE Data compression conference March 2003.

[10] S. Lonardi, and W. Szpankowski, “Error resilient LZ’77 Scheme and analysis,” In Proceeding IEEE international symposium on information Theory, USA 2004.

[11] “Improving LZ77 Data compression using Bit Recycling. In international Symposium on information Tehory and its Application”, Seoul Korea, 2005.

[12] S. Lonardi, and W. Szpankowski, “Error resilient LZ’77 Data compression: Algorithms, Analysis and Experiment”, Chigago USA 2006.

[13] Ziv, J, and Lempel, A. “A Universal Algorithm for Sequential Data Compression,” IEEE Trans. on Inf. Theory IT-23, pp. 337-343, May 1977.

[14] T.A. Welch,” A technique for high Performance data compression", Computer, Vo1.17. No. 6, pp 8-19, June 1984.

[15] Mohammed Al-laham1 & Ibrahiem M. M. El Emary2 . “Comparative Study between Various Algorithms of Data Compression Techniques”, IJCSNS, April 2007.

[16] Balwant Singh Raghuwanshi, Piyush Kumar Shukla, Pradeep Rusiya, Deepak Agrawal, Lata Chhablani, “Multiple Subgroup Data Compression Technique Based On Huffman Coding” *CICSN*, First International Conference on Computational Intelligence, Communication Systems and Networks, 23-25, July 2009.

[17] I. E. Bocharova, R. Johannesson, andB. D. Kudryashov, “Low state complexity block codes via convolutional codes,” *IEEE Trans. Inf.Theory*, Vol. 50, No. 9, pp.2022–2030, Sep. 2004

[18] R. Dehariya, P. Shukla, M. Bargadie, S. Kumar, “Switching code data compression technique using an Adaptive Huffman coding”, Mathematics and Computers In Science And Engineering archive Proceedings of the American Conference on Applied Mathematics table of contents Cambridge, Massachusetts, pp. 431-436, 2008.

A Block Cipher Involving a Key Applied on Both the Sides of the Plain Text

V. U. K. Sastry[1], D. S. R. Murthy[2], S. Durga Bhavani[3]

[1]Dept. of Computer Science & Engg., SNIST,
Hyderabad, India,
vuksastry@rediffmail.com

[2]Dept. of Information Technology, SNIST,
Hyderabad, India,
dsrmurthy.1406@gmail.com

[3]School of Information Technology, JNTUH,
Hyderabad, India,
sdurga.bhavani@gmail.com

Abstract: ***In this paper, we have modified the Hill Cipher by introducing an iterative procedure, which includes multiplication with the key matrix on both the sides of the Plain text matrix, mixing of the plain text by using a function called Mix ( ), and XORing of the plain text matrix and the key matrix, at every stage of iteration. The cryptanalysis carried out in this paper, clearly indicates that the cipher is a very strong one and it cannot be broken by any cryptanalytic attack.***

Keywords: Cipher, Modular arithmetic inverse, Plain text, Cipher text, Key.

1. Introduction

In a pioneering paper, Hill [1] developed a block cipher by using the modular arithmetic inverse of a square matrix. In this, the 26 characters of English alphabet are represented by the numbers 0 to 25, and the encryption is carried out by using the relation

$$\mathbf{C = K\,P \bmod 26}, \qquad (1.1)$$

where **P** is the plain text column vector, **C** the cipher text column vector, and **K** is the key matrix. In the process of decryption, the plain text **P** is obtained by using the relation

$$\mathbf{P = K^{-1}\,C \bmod 26}, \qquad (1.2)$$

where K^{-1} is the modular arithmetic inverse of K governed by the relation

$$\mathbf{K\,K^{-1} \bmod 26 = I}. \qquad (1.3)$$

However, it is established in the literature [1] that this cipher can be broken by the known plain text attack as we can form a relation of the form

$$\mathbf{Y = K\,X \bmod 26}, \qquad (1.4)$$

where X is a square matrix containing the plain text column vectors and Y is the matrix containing the corresponding cipher text column vectors, and the modular arithmetic inverse of X can be found so that it leads to the relation

$$\mathbf{K = (Y\,X^{-1}) \bmod 26}. \qquad (1.5)$$

In a recent paper, Sastry and Janaki [2] have developed a systematic procedure for the modular arithmetic inverse of a square matrix. Then Sastry et al. [3]–[5] have modified the Hill cipher by introducing iteration, transposition, and / or permutation. In all these investigations, it has been seen that the process of iteration does not allow the formation of a direct relation of the form (1.4) and hence, the known plain text attack is not possible.

In the present paper, our objective is to modify the Hill cipher, so that the known plain text attack is impossible. Here, we use the relation

$$\mathbf{C = (K\,P\,K) \bmod 2}, \qquad (1.6)$$

for encryption, where P is a square matrix of the plain text. The process of decryption is governed by the relation

$$\mathbf{P = (K^{-1}\,C\,K^{-1}) \bmod 2}. \qquad (1.7)$$

It may be noted here that the presence of K, after P in (1.6), will certainly rule out the known plain text attack and strengthen the cipher under consideration.

In section 2, we have presented the development of the cipher. We have illustrated the cipher in section 3, and discussed the cryptanalysis and the avalanche effect in section 4. Finally, we have presented some numerical computations and have drawn conclusions in section 5.

2. Development of the cipher

Consider a plain text P which can be represented in the form of a square matrix given by

$$\mathbf{P = [P_{ij}]}, \quad i = 1 \text{ to } n,\ j = 1 \text{ to } n, \qquad (2.1)$$

where each P_{ij} is either 0 or 1.

Let us choose a key k. Let it be represented in the form of a matrix given by

$$\mathbf{K = [K_{ij}]}, \quad i = 1 \text{ to } n,\ j = 1 \text{ to } n, \qquad (2.2)$$

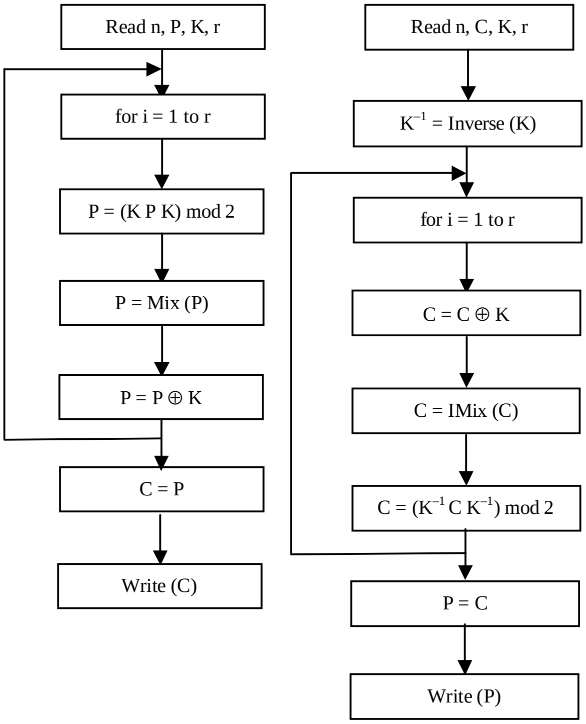
where each K_{ij} is a binary number.

Let

$$\mathbf{C = [C_{ij}]}, \quad i = 1 \text{ to } n,\ j = 1 \text{ to } n \qquad (2.3)$$

be the corresponding cipher text matrix.

The process of encryption and the process of decryption adopted in this analysis are given in Figure 1.



(a) Process of Encryption (b) Process of Decryption

Figure 1. Schematic diagram of the cipher

In the process of encryption, before the plain text P transforms to the cipher text C, we have used an iterative procedure. The iteration is carried out for r times. In this, the relations

$\mathbf{P = (K\,P\,K) \bmod 2,}$ (2.4)

$\mathbf{P = Mix\,(P),}$ (2.5)

and $\mathbf{P = P \oplus K}$ (2.6)

are introduced for achieving diffusion and confusion, so that they will add strength to the cipher.

Let us consider **Mix (P)**. In this P is containing n^2 binary numbers. They can be written in the form of a string given by

$P_{11}, P_{12}, \ldots, P_{1n}, P_{21}, P_{22}, \ldots, P_{2n}, \ldots, P_{n1}, P_{n2}, \ldots, P_{nn}$.

If n^2 is divisible by 8, then the above string can be divided into a set of substrings, wherein the length of each substring is 8. Then we focus our attention on the first 8 substrings. We place the first bits of these 8 binary substrings, in order, at one place and form a new binary substring. Similarly, we assemble the second 8 bits and form the second binary substring. Following the same procedure, we can get six more binary substrings in the same manner. Continuing in the same way, we exhaust all the binary substrings obtained from the plain text.

However, if n^2 is not divisible by 8, then we consider the remnant of the string, and divide it into two halves. Then we mix these two halves by placing the first bit of the second half, just after the first bit of the first half, the second bit of the second half, next to the second bit of the first half, etc. Thus we get a new binary substring corresponding to the remaining string. This completes the process of mixing.

In the process of decryption, the function IMix represents the reverse process of Mix.

In what follows, we present the algorithms for encryption, and decryption. We also provide an algorithm for finding the modular arithmetic inverse of a square matrix.

Algorithm for Encryption

```
1.  Read n, P, K, r
2.  $K^{-1}$ = Inverse (K)
3.  for i = 1 to r
    {
     P = (K P K) mod 2
     P = Mix (P)
     P = P $\oplus$ K
    }
4.  C = P
5.  Write (C)
```

Algorithm for Decryption

```
1.  Read n, C, K, r
2.  $K^{-1}$ = Inverse (K)
3.  for i = 1 to r
    {
     C = C $\oplus$ K
     C = IMix (C)
     C = ($K^{-1}$ C $K^{-1}$) mod 2
    }
4.  P = C
5.  Write (P)
```

Algorithm for Inverse (K)

```
// The arithmetic inverse ($A^{-1}$), and the determinant of the
    matrix ($\Delta$) are obtained by Gauss reduction method.
1.  A = K, N = 2
2.  $A^{-1}$ = [$A_{ji}$] / $\Delta$,  i = 1 to n, j = 1 to n
        //$A_{ji}$ are the cofactors of $a_{ij}$, where $a_{ij}$ are elements
          of A, and $\Delta$ is the determinant of A
3.  for i = 1 to n
    {
     if ((i $\Delta$) mod N = 1)
        d = i;
     break;
    }
4.  B = [d $A_{ji}$] mod N
     // B is the modular arithmetic inverse of A
```

3. Illustration of the cipher

Let us consider the following plain text.

Dear Friend! Terrorism and recession are the terrible problems of the entire Globe. We have to unite and find a solution to these tyrannical problems. Let us put-in our whole hearted effort. I need not emphasize that our country is all the while to achieve peace in all parts of the universe. Please agree to my proposal (3.1)

Let us focus our attention on the first 32 characters of the above plain text which is given by

Dear Friend! Terrorism and reces. (3.2)

On using the EBCDIC code, the plain text under consideration can be written in Hexadecimal notation as follows:

C4 85 81 99 40 C6 99 89 85 95 84 4F 40 E3 85 99
99 96 99 89 A2 94 40 81 95 84 40 99 85 83 85 A2
(3.3)

On placing two numbers in each row, that is, C4 85 in the first row, 81 99 in the second row, etc., the plain text matrix P can be written in the binary form as

$$P = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} \qquad (3.4)$$

Let us choose a key k in the decimal form as

131 31 18 59 254 126 113 97 127 167 76 116 111 159 245 159
175 50 236 107 235 74 47 20 190 80 242 139 175 164 187 158
(3.5)

Then the Key matrix K be obtained as

$$K = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0
\end{bmatrix} \qquad (3.6)$$

On using the algorithm given in Section 2, the modular arithmetic inverse of K can be obtained as

$$K^{-1} = \begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0
\end{bmatrix} \qquad (3.7)$$

On using (3.6) and (3.7), it can be readily shown that

$$\mathbf{K\,K^{-1} \bmod 2 = K^{-1}K \bmod 2 = I}. \qquad (3.8)$$

On applying the encryption algorithm, described in Section 2, we get the cipher text C in the form

$$C = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1
\end{bmatrix} \qquad (3.9)$$

On using (3.7) and (3.9), and applying the decryption algorithm described in section 2, we get the Plain text P, which is the same as (3.4).

Let us now study the avalanche effect. To this end, we focus our attention on (3.2), and modify the 11th character ‘d’ to ‘e’. Then the plain text changes only in one binary bit as the EBCDIC codes of d and e are 84 and 85 respectively. On using the encryption algorithm, we get the cipher text C corresponding to the modified plain text in the form

$$C = \begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1
\end{bmatrix} \qquad (3.10)$$

On comparing (3.9) and (3.10), we find that the two cipher texts differ in 124 bits, out of 256 bits, which is quite significant.

Now let us change the key by one binary bit. To this end we replace the 3rd element 18 of the key k by 19. Then on using the original plain text given by (3.4), we get C in the form

$$C = \begin{bmatrix} 1&1&1&1&0&1&0&0&0&1&1&1&1&0&0&1 \\ 1&0&0&1&0&0&0&1&1&0&1&0&1&0&0&0 \\ 0&0&0&0&0&0&1&1&1&0&1&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1&1&0&0&0&1&0&0&1 \\ 1&1&0&0&1&0&1&0&0&0&0&1&0&0&0&1 \\ 1&1&1&0&1&1&1&0&1&0&1&0&0&0&1&1 \\ 1&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0 \\ &1&0&1&0&1&0&1&1&1&1&1&1&1&0&1&1 \\ 0&0&1&1&1&1&0&0&1&1&0&1&0&1&0&1 \\ 0&1&0&1&0&1&1&1&0&1&1&0&0&0&1&0 \\ 0&1&0&1&1&0&0&1&1&1&0&1&0&0&1&1 \\ 0&0&1&1&0&0&1&1&1&1&0&1&1&0&1&1 \\ 0&0&0&1&0&1&0&1&0&0&0&1&1&1&1&0 \\ 1&1&0&1&0&0&0&1&1&1&1&1&0&0&1&1 \\ 1&0&0&1&1&0&0&0&0&0&1&0&1&0&0&1 \\ 0&0&0&1&0&0&0&1&1&1&1&0&1&0&0&0 \end{bmatrix} \qquad (3.11)$$

On comparing (3.9) and (3.11), we find that the cipher texts differ in 117 bits, out of 256 bits.

From the above analysis, we notice that the avalanche effect is quite pronounced and hence the cipher is a strong one.

4. Cryptanalysis

The different types of attacks for breaking a cipher are:

(1) Cipher text only attack,
(2) Known plain text attack,
(3) Chosen plain text attack
(4) Chosen cipher text attack.

When the cipher text is known to us, we can determine the plain text, if the key is known. As the key contains 32 decimal numbers, the key space is of size

$$2^{256} \simeq (10^3)^{25.6} = 10^{76.8}.$$

Hence, in this case, the cipher cannot be broken by applying brute force approach.

In the case of the Hill cipher, we know that, it can be broken by the known plain text attack, as we can form a direct relation between C and P. But in the present modification, which involves K on both the sides of P, and the process of iteration together with Mix function and XOR operation, we cannot get a direct relation between C and P. Hence, this cipher cannot be broken by the known plain text attack.

Obviously, no special choice of plain text or cipher text does enable us to break the cipher.

5. Computations and Conclusions

In this paper, we have modified the Hill cipher, which depends upon the single relation

C = (K P) mod 26, (5.1)

by introducing an iterative scheme, which involves the relations

P = (K P K) mod 2, (5.2)
P = Mix (P), (5.3)
and **P = P ⊕ K,** (5.4)
followed by **C = P**. (5.5)

In this modification, the K on the right side of the P and the iterative process, which includes a Mix function, and a XOR operation, modifies P very significantly, and no direct relation can be obtained between the cipher text C and the original plain text P. Thus the cipher cannot be broken by the known plain text attack.

By decomposing the entire plain text given by (3.1) into blocks, wherein each block is of size 32 characters, the corresponding cipher text can be written in hexadecimal notation in the form

```
97 72 5A FC B8 F2 89 BE CE A5 2D 77 88 3D AE 8B
68 45 94 03 CA 33 59 13 A4 2F 6F 31 0B 29 D9 D7
5F 37 AC 62 92 19 23 44 8A DA FE 06 E2 78 0E CE
67 F5 7A 5A 57 8A 3A 74 49 C7 5B 9B 0B 5B FA 33
FB D5 DF 53 CE 6E E6 C5 F0 03 F1 AE D6 23 3F DE
E7 DC 96 0E 6F 68 AB E7 4B 20 F5 4E 70 17 5A 9B
A1 B1 FF 4A 83 A8 EE E2 A4 BC 06 B1 93 72 2E D9
33 20 B4 33 F1 11 21 CE 0F BD 0C D0 85 F5 F0 03
1A 26 6A 4E 55 B4 02 EC 1C AB 4C DD 9C 43 67 47
28 DB D4 FA 18 33 FB 2F 01 F8 E2 23 86 39 B7 5E
A0 57 D5 EB 87 B1 08 61 FF 4A 00 51 88 47 BB 44
D3 BE 4D A3 BE 43 8E 14 5B 69 EC 11 AE 2D 29 5A
30 D7 F8 C6 A1 4C F5 32 EF 54 1E DF 53 0E 71 F4
0C B9 EA 85 DE 2F 10 C6 DB 29 61 A8 0F 36 83 3D
23 3B 44 2C 53 A7 9E 16 AF FE F4 F8 87 FB 3B E3
DB 53 D1 98 10 4A F4 0E 29 02 66 A5 F5 8B 5D C5
AB 67 5A 0B FA EE D7 12 C9 C6 CA 72 27 B5 08 FD
60 26 F9 EE 1E B0 BA 57 01 DE 0A 99 AF A9 64 77
90 BE F5 06 09 CD AF 70 CC 42 97 35 FE 60 4B 45
F4 F7 00 7D 3E 89 E8 22 80 DD 88 72 AE 7A 4F 73
```

In this analysis, the length of the plain text block is 256 bits and the length of the key is 256 bits. As the cryptanalysis clearly indicates, this cipher is a strong one and it cannot be broken by any cryptanalytic attack. This analysis can be extended to a block of any size by using the concept of interlacing [5].

References

[1] William Stallings, *Cryptography and Network Security*, Principles and Practice, Third Edition, Pearson, 2003.

[2] V. U. K. Sastry, V. Janaki, "*On the Modular Arithmetic Inverse in the Cryptology of Hill Cipher*", Proceedings of North American Technology and Business Conference, Canada, Sep. 2005.

[3] V. U. K. Sastry, S. Udaya Kumar, A. Vinaya Babu, "*A Large Block Cipher using Modular Arithmetic Inverse of a Key Matrix and Mixing of the Key Matrix and the Plaintext*", Journal of Computer Science 2 (9), pp. 698 – 703, 2006.

[4] V. U. K. Sastry, V. Janaki, "*A Block Cipher Using Linear Congruences*", Journal of Computer Science 3(7), pp. 556 – 561, 2007.

[5] V. U. K. Sastry, V. Janaki, "*A Modified Hill Cipher with Multiple Keys*", International Journal of Computational Science, Vol. 2, No. 6, pp. 815 – 826, Dec. 2008.

Identity Application Fraud Detection using Web Mining and Rule-based Decision Tree

Amany Abdelhalim[1] and Issa Traore[2]

[1]Department of Electrical and Computer Engineering
University of Victoria, P.O. Box 3055 STN CSC,
Victoria, B.C., V8W 3P6, Canada
Email: amany@ece.uvic.ca

[2]Department of Electrical and Computer Engineering
University of Victoria, P.O. Box 3055 STN CSC,
Victoria, B.C., V8W 3P6, Canada
Email: itraore@ece.uvic.ca

Abstract: ***Identity fraud is becoming a growing concern for most government and private institutions. In the literature, identity frauds are categorized into two classes, namely application fraud and behavioural (or transactional) fraud. Most of the previous works in the area of identity fraud prevention and detection have focused primarily on credit transactional frauds. The work described in this paper is one of the very few works that focus on application fraud detection. We present an unsupervised framework to detect fraudulent applications for identity certificates by extracting identity patterns from the web, and crossing these patterns with information contained in the application forms in order to detect inconsistencies or anomalies. The outcome of this process is submitted to a decision tree classifier generated on the fly from a rule base which is derived from heuristics and expert knowledge, and updated as more information is obtained on fraudulent behaviour. We evaluate the proposed framework by collecting real identity information online and generating synthetic fraud cases.***

Keywords: Application fraud, Fraud detection, Anomaly detection, Web mining, Rule-based Decision Tree.

1. Introduction

Identity fraud is spreading fast and causing more and more damages both financially and sociologically. Identity fraud occurs when a criminal impersonates another individual by taking on that person's identity or by creating a fake identity for whatever reason [1] and [2].

Identity frauds can be categorized into two different types: transaction frauds and application frauds. Application fraud occurs when an individual or an organization applies for an identity certificate (e.g., passport, credit card etc.) using someone else’s identity. Transaction fraud, also known as behavioral fraud, occurs when an identity thief performs some operations or transactions using fake or stolen identity. Most of the research in identity fraud detection has focused so far on credit transactional fraud detection. Limited attention has been paid to application fraud detection, where only few papers have been published so far. Application fraud detection, however, is an important aspect of any sound and global strategy to combat identity fraud. Application fraud detection is a proactive measure that allows early screening of fraudsters, contributing as a result to cutting down significantly the effort and resources required to detect fraudulent transactions [3].

Application frauds share many of the characteristics of transaction frauds, including the large amount of data processed, the need to make an acceptance or rejection decision in a short time span, and the significant imbalance between normal and fraudulent data (fraud cases represent only a very small fraction of performed operations). These issues have widely been studied in the transaction fraud literature. The open issues concern the differences between application and transaction frauds.

Transactional operations benefit typically from significant historical records collected over time for a single person. Many Artificial Intelligence (AI) techniques can be used to process past usage history and provide fairly accurate fraud detection results. In contrast historical information related to application operations for a specific person is in general very limited if non existent. For instance, there is a significant delay between consecutive applications for passport renewal in most countries; in some countries it takes between 5 to 10 years before having to renew a passport. As a result the background historical data per individual is very limited. Furthermore the application file contains only sparse identity attributes, which by themselves might not be enough to make a proper decision. So application fraud detection requires specific analysis techniques which can compensate for the weak characteristics of the available data.

Another key difference between transaction and application fraud data is the breadth of the identity information covered. While as discussed above, transaction fraud data involve significant depth (at least from historical standpoint), application fraud data have wider scope. Application forms may potentially be linked to a wider variety of data sources than what typical transactional operations could enjoy. For instance, information contained in a passport application may be crossed with collateral information in many other databases like vital statistics, death records, social security database, birth records etc. In contrast transactional operations typically cover only a limited number of different data sources maintained internally by the issuer or available externally. For instance, for legal and competitive reasons, credit card transactions can be cross-checked against only few external databases such as credit bureaus.

An alternative identity information source that transcends the above restriction is the web. The web is actually a federation of many different identity information data sources. Using the web, it may be possible to cross-check application forms with data from various collateral identity information sources; even though such information might be sparse or useful information could be missing.

The web is one of the richest and diverse sources of identity information. But at the same time it is one of the most challenging to deal with because of the unstructured nature of the data involved. To our knowledge none of the application fraud detection frameworks proposed so far has explored the strength of such data source. All of them use traditional offline or private data sources. In many ways, the Internet serves as a key vehicle for identity fraud. The Internet represents an appealing place for fraudsters to collect a host of personal and financial data related to many innocent users. Using the collected data they can impersonate the users and commit fraudulent activities using stolen or fake identities. Mining Internet data for fraudulent purposes using a search engine is commonly referred to as *black hat Google hacking.*

In this work, we propose an application fraud detection framework that consists of two main components: an online identity mining module and a fraud detector. Our proposed online identity mining scheme is based on *white hat Google hacking*, in which identity information is collected through online search, targeting a specific individual (i.e. the applicant). The fraud detector is an intelligent unsupervised decision model that analyzes extracted online identity information related to the applicant and crosses such information with information contained in the application form, in order to detect and report possible inconsistencies or anomalies. We designed our fraud detector using a rule-based decision tree technique fed with a set of simple heuristics that define general and common understanding of the notion of fraudulent and normal behaviours.

Although various machine learning techniques (supervised or unsupervised) may be used in designing such kind of detector, the lack of genuine fraud data tends to hinder such process. A common challenge of data-mining based fraud detection research is the lack of publicly available real data for model building and evaluation. For privacy and competitive reasons, organizations are reluctant to release data related to fraudulent activities.

In order to compensate for the fact that labels may not be readily available, ideally such techniques should be unsupervised. Although many unsupervised detection frameworks have so far been proposed for transaction fraud detection, none of them apply specifically to application fraud detection. To our knowledge, the application fraud detection techniques proposed so far in the research literature are either supervised or semi-supervised.

The solution commonly adopted in the industry and in the literature to address this issue is to encode expert knowledge and past knowledge of fraudulent behavior into rule bases. Likewise, the application fraud detection techniques used in commercial applications and in the research literature include rule-based matching of credit application and credit history, fraud matching using black lists (based on previous fraudulent applications), and supervised model based on labeled data. However, due to the fast pace at which new fraud methods are created and used by fraudsters, the rule bases are submitted to constant changes and usually tend to grow at an accelerated rhythm, quickly reaching unmanageable size. This might not be conducive to timely decision, which is required in many business environments. In this case, a decision tree represents an effective alternative to rule-base reasoning for a quicker decision. This is because in order to be able to make a decision for some situation we need to decide the shortest and most efficient order in which tests should be evaluated. In that case a decision structure (e.g. decision tree) is much quicker than a rule engine to reach a decision. In the decision tree the order of checking conditions and executing actions is immediately noticeable. Second, conditions and actions of decision trees are found on some branches but not on others. Those conditions and actions that are critical are connected directly to other conditions and actions, whereas those conditions that do not matter are absent. In other words it does not have to be symmetrical. Third, decision trees are more readily understood by others in the organization than a set of rules. Consequently, they are more appropriate as a communication tool.

Decision trees can be an effective tool for guiding a decision process as long as no changes occur in the conditions or dataset used to create the decision tree. When the conditions change, as this happens constantly with fraud cases, restructuring the decision tree becomes a desirable task. It is difficult, however, to manipulate or restructure decision trees. This is because a decision tree is a procedural knowledge representation, which imposes an evaluation order on the attributes. In contrast, a declarative representation, such as a set of decision rules is much easier to modify and adapt to different situations than a procedural one. This easiness is due to the absence of constraints on the order of evaluating the rules [4].

Rule-based decision tree techniques bridge the divide between rule-base systems and decision trees, by allowing the on-demand creation of a short and accurate decision tree from a stable or dynamically changing set of rules. On one hand decision tree paradigm easily allows changes to the conditions (when needed) by modifying the rules rather than the decision tree itself. On the other hand the paradigm takes advantage of the structure of the decision tree to organize the rules in a concise and efficient way required to take the best decision. So knowledge can be stored in a declarative rule form and then be transformed (on the fly) into a decision tree only when needed for a decision making situation [4] using a rule-based decision tree technique. We use *RBDT-1*, a rule-based decision tree technique recently proposed in [5] to design our fraud detector. *RBDT-1* has been shown to be more effective in terms of tree complexity than other existing similar techniques [6].

We evaluate the proposed framework using a database of real online identity information collected through white hat

Google hacking combined with a synthetic sample of fraud data that we created.

The rest of the paper is structured as follows. In Section 2, we summarize and discuss related work. In Section 3, we present our approach and give an overview of the general architecture of the system. In Section 4, we present our online identity information retrieval scheme. In Section 5, we present our identity fraud detection scheme. In Section 6, we conduct the evaluation of the proposed framework. Finally in Section 7, we make some concluding remarks.

2. Related Work

Although there is a large amount of published works on identity fraud detection, only a few of these works focus specifically on application fraud detection. We review in this section representative samples of these works.

Wang *et al.* developed a general framework to analyze identity theft in the context of integrated multiparty perspectives [2]. Although their work is very important, it is mostly conceptual and focuses on identifying the stakeholders involved in identity fraud detection and protection, and on characterizing the interactions between them. According to their framework, there are five main stakeholders: identity owner, identity protector, identity issuer, identity checker, and identity thief. Although we use some of the concepts defined in their framework, our work is different from their work in the sense that we focus primarily on studying, implementing and evaluating a concrete and practical identity checker.

Burns and Stanley in [7] discuss the techniques used by credit card issuers to screen credit applications. Card issuers use multiple data sources (e.g., credit bureaus) to confirm the information listed in an application form. They monitor relevant databases for any changes to the consumer's credit records, personal address or phone number, which often give the earliest indication of identity theft. They also calculate spatial information such as the distance between the phone number and address presented in the application to determine if they originate from the same area code or not. Applications which are submitted from areas where a lot of fraud cases appeared before are also reviewed thoroughly.

Cross-referencing new applications with similar information from other databases is a common characteristic of most of the application fraud detection approaches proposed so far in the literature, including [8], [9] and [10].

Wheeler and Aitken in [8] applied case-based reasoning for credit card application fraud detection. The proposed system was used as reinforcement for an existing rule-based (RB) fraud detector. The input data to the system consists of pairs of database records, consisting in one hand of the application and on the other hand of fraud evidence produced by the RB system. The goal is to reduce the fraud investigations by combining the diagnosis of multiple algorithms in producing the final decision. The proposed case based reasoning framework consists of two decision-making modules, in charge of case retrieval and diagnosis, respectively. The retrieval component utilizes a weighting matrix and nearest neighbor matching to identify and extract appropriate cases to be used in the final diagnosis for fraud. The decision component utilizes a set of algorithms (i.e. probabilistic curve selection, best match algorithm, negative selection algorithm, density selection algorithm, and default goal) to analyze the retrieved cases and attempt to reach a final diagnosis. The results of the system showed that the performance of each algorithm employed in the fraud diagnosis process differs depending on the nature of the fraud case presented. In our work we also use application cross-referencing to capture similarities. However, our fraud detection algorithm does not need labels to make fraud or non-fraud decisions. While in the work of Wheeler and Aitken, the evidence has to be produced and tagged as fraud or non-fraud by a separate system, our proposed system uses unlabelled data in its decision-making process.

Phua *et al.* in [9] proposed a technique for detecting application fraud based on implicit links between new and previous applications. Using a communal suspicion scoring scheme, they classify a new application as belonging to one of three lists, a black list, a white list or an anomalous list. They performed the classification by finding a match between information contained in a new application and information from an existing application in one of the lists. Both the information in the new application and the corresponding linked application were represented by an attribute vector. The suspicious score is the summation of each pair-wise attributes which is expressed by either exact matching or fuzzy matching. They also assigned weights to the attributes depending on their nature. The proposed technique is similar to the weight matrix proposed in [8], except that in [9], in addition to basing their calculation of suspicion scores on matching attributes, they take into consideration temporal and spatial differences between the matching applications. A key difference between this approach and ours is that it requires labeling the data. Despite this important difference, we evaluate our work by mainly comparing it to the approach proposed by Phua *et al.*, as explained later.

In [10], a system that detects subscription fraud in fixed telecommunications was proposed. The system consisted of two main modules, a classification module and a prediction module. The purpose of the prediction module was to detect fraudulent customers when they attempt to subscribe to a fixed telecommunication service. To investigate the application for signs of fraud, the prediction module crosses the information available in the new application with information available in the account database. This allows linking the new application with existing fraudulent accounts. Another source of information that was used was a public database that lists situations of insolvency mostly related to banks and department stores. The prediction module consisted of a multi-layer feed-forward neural network. The output units indicated one of two decisions for an application; either fraudulent or legitimate. A dataset of subscription examples was labeled and split into the following three sets: a training set, a validation set, and a testing set. From their experiments they outlined that the prediction module was able to identify only 56.2% of the true fraud cases while screening 3.5% of all the subscribers in the testing set. Like in [8] and [9], the proposed framework requires using labeled data and some form of supervision. Many criticisms can be made about using labeled data for fraud detection such as the limited efficiency of processing such data in event-driven environment, the cost, difficulty, and length of time needed to obtain such data, and the fact that the class labels can be inaccurate. In our work, we do not use or assume knowledge of existing fraudulent applications in the screening process. Instead, our proposed fraud detector processes unlabelled data from the identity information source.

In the Identity Angel approach proposed by Sweeney, identity information is extracted from the Web [11] and [12]. The goal of the Identity Angel project is to locate online resumes that contain sufficient information for a criminal to acquire a new credit card in the name of the owner of the resume, and then to notify the corresponding subject by sending him an email, encouraging him to remove such sensitive information.

Identity Angel is implemented using a Java program that uses filtered search and the Google API to identify resumes, and uses entity detectors for SSNs, dates of birth, and email addresses to extract information from online resumes. While the identity information retrieval technique used by Sweeney is closely related to ours, an important difference is that the search space used in the identity angel project is limited to a specific kind of online documents, which are online resumes. In our case, our search space is the entire web and targets any kind of documents, which gives a wider scope for our knowledge source. Furthermore, the identity angel tool focuses only on information retrieval and does not implement any formal fraud detection process or algorithm.

3. Conceptual Framework and Architecture

In this section, we describe basic identity concepts and give an outline of our identity fraud detection framework design.

3.1 Identity Concepts

Identity can be defined "as one or more pieces of information that cause others to believe they know who someone is" [13]. We divide such pieces of information into two categories: basic identity information and specific identity information. Basic identity information simply refers to the first name and last name of an individual. Specific identity information refers to additional identifying information other than the basic identity information, such as birth date, social security number, address, credit card number, etc. A targeted search strategy typically consists of using basic identity information to obtain specific identity information for a particular individual.

In [2], the notion of identity certificate is introduced. An identity owner applies for an identity certificate for various purposes in his life, either administrative or business related. Examples of identity certificates include passport, social security card, driver’s license, health card, credit card, digital (security) certificate, etc. Identity issuers, represented by trusted government or private institutions, deliver identity certificates.

An identity certificate usually includes at least one or several of the following pieces of identity information: certificate number, owner’s information, the purpose of the certificate (social security or credit card), issuer’s information, validity time period, issuer’s certification, and owner security information such as his signature or the security number used for credit card. An identity owner is usually characterized by providing at least one or several of the following pieces of identity information: the first and last names, the parents’ names with a particular emphasis on mother’s maiden name, the address, and the date and place of birth.

Although identity issuers have various and more or less sophisticated ways to check the validity of identity certificates, identity certificates still represent the main vehicle used to conduct identity fraud. The issuing and use of identity certificates rely on a chain of trust. For instance, the issuing of a credit card relies on the social security card, which in its turn relies on the passport, which again relies on the birth certificate. This chain of trust can be broken, when a fraudster is able to steal one’s identity or to fake a new one. To obtain an identity certificate an individual submits an *application* to a certificate issuer providing required identification information. In doing so, she makes an *identity claim.* So, an *identity claim* occurs when an individual declares a specific identity, for instance, on an official document like a passport application, or a business document like a credit card application.

3.2 General Architecture

Our application fraud detection framework consists of three main modules as illustrated by Figure 1: the identity information source, the identity information retrieval engine, and a fraud detector. The identity information source can be a single or a combination of several identity data sources such as credit bureaus databases, previous applications databases, vital statistics, or the web.

The identity information retrieval engine queries the identity information source by targeting the identity applicant, and feeds the targeted search results into the identity fraud detector. The identity fraud detector crosses and analyzes, using heuristics, the search results with background identity information fetched from the application file (e.g., credit card application) being checked.

The identity information retrieved for a particular individual consists of a collection of identity information patterns. Each pattern may consist of a sequence of identity information pieces such as address, credit card number, mother maiden name, social security number etc. Resulting from a targeted search process, the patterns may belong to different individuals that simply happen to share the same name. So it is necessary to develop a strategy to sort and narrow down the returned patterns for the target individual. The sorted information can then be analyzed to detect possible identity fraud.

Identity fraud detection will consist of screening a particular identity claim for possible inconsistencies with the information in the search results. The system will extract the identity information for the individual found in the search results and check the results against the identity claim, reporting any discrepancy as an anomaly. This may either lead to a rejection of the application or to further investigations with some follow-up questions.

We will describe in subsequent sections the identity information retrieval engine and the identity fraud detector.

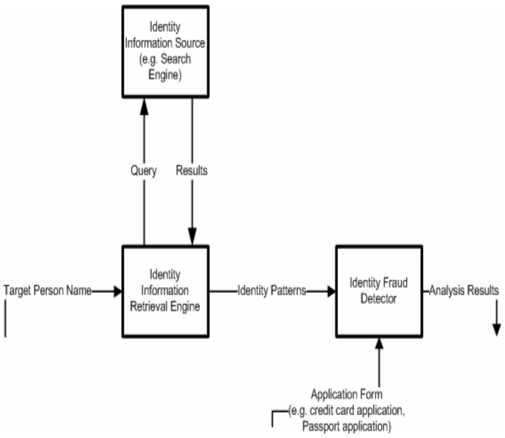

Identity Information Source (e.g. Search Engine)
Query
Results
Target Person Name
Identity Information Retrieval Engine
Identity Patterns
Identity Fraud Detector
Analysis Results
Application Form (e.g. credit card application, Passport application)

Figure 1. Fraud Detection Framework

4. Web-based Identity Information Retrieval Engine

The design of the identity information retrieval engine depends on the identity information source. Unlike with other identity information sources, identity information retrieval on the web involves a lot of challenges because of the scale, complexity, and diversity of the information provided. On the other hand, the web represents a powerful source of identity information that can be used effectively in combating identity fraud. We describe in this section the challenges and algorithms involved in the design of our web-based identity information retrieval engine.

4.1 Design challenges and strategies

Although many tools are readily available for information retrieval, identity information retrieval based on targeted search is a challenging task. Because of the huge amount of information available online, locating and sorting identity information related to a specific individual is a daunting task. We discuss in the following paragraphs some of the challenges involved in this task.

A key challenge is that a wide variety of document formats are used on the Internet (e.g., PDF, Doc, Excel, or Html). Some of these formats are not directly searchable, and require some form of pre-processing. Also our search space is not limited to a specific type of documents. As a matter of fact searching and extracting identity information from a document, which has only identity values for the target name such as the curriculum vitae is easier than doing the same with a document that could have more than one person's identity information such as a company employee list. For the latter, the challenge is to avoid extracting identity information for someone different from our target.

Furthermore it is difficult to establish the existence of values corresponding to the identity keywords located in the documents. The reason for such difficulty is that although some of the identity keywords targeted in our search have a fixed number of characters, such as social security number or telephone number, no standard format is used to express corresponding values in the documents on the Internet. So, it is necessary to design a general value filter that is capable of finding and extracting identity values based on the common ways that people usually tend to express them online. One of the proposed solutions is to limit the search area in the document starting from the target name and ending at the next existing name.

Another challenge is that some of our identity keywords have homonyms. In many cases, sorting information pertaining to different homonyms is challenging. In the literature, one of the main approaches used to handle homonyms consists of associating semantic information with the keyword. The semantic-based approach, however, does not make sense when the homonyms are proper nouns. Since we are dealing, in our work, with targeted search, homonyms based on proper nouns are very important. We handle this kind of homonyms using our identity profile derivation strategy, outlined in Section 5, which allows sorting and separating identity information belonging to different individuals who happen to be homonyms.

Besides that, each of the identity information pieces, searched for, has synonyms. Because our search space is the Internet, there is no standard word for expressing these keywords. So we have to come up with all the possible synonyms for each of the identity keywords and expand the search parameters in the identity keyword filter to include those synonyms.

Algorithm: *The Identity Information Detector IID*

Input: The target person's name *n*.

Output: A list of documents containing at least one value for any of the keywords detected, along with those keywords and their values.

Uses: Set of search keywords *Keys* and the web.

```
Steps:
let $D = \emptyset$, a set that will contain all the documents returned from the search.
let $D_t = \emptyset$, a set that will contain all the documents returned from the search after converting them into text.
  for each $s \in$ Keys do:
      D= SearchEngineApi (s+n)
  endfor

  for each $d_i \in$ D do:
      $D_t$= ConvertToText($d_i$) $\cup$ $D_t$
  endfor

  for each $d_i \in D_t$ do:
      ArrayofKeys= KeywordFilter ($d_i$)
      KeyValue = ValueFilter ($d_t$, ArrayofKeys)
      if KeyValue $\neq \emptyset$
            result[i][1] = $d_i$
            result [i][2] = KeyValue
      endif
  endfor
 return result
```

Figure 2. Identity Information Detection (IID) Algorithm

4.2 Search algorithm

In order to illustrate our concepts, we limit (in our proof of concept) our search keywords to only credit card and social security information, which correspond to two of the most popular and sensitive identity certificates available online. We think that this does not affect in anyway the generality of our framework.

Our identity information retrieval scheme will use targeted search based on the following specific identity information:

- Social security number
- Date of birth
- Home address
- Home telephone number
- Mother maiden name
- Credit card number
- Credit card expiry date
- Credit card type
- Credit card security number

Table 1: List of Keywords for Automated Identity Information Retrieval

Identity Information	Search Keywords
Social security number	[ssn| social security number| social security no| social security #| ssn#| ssnum| ssno]
Date of birth	[date of birth| born| dob| birthplace and date| d.o.b| birthdate]
Home address	[address| add| home| home address]
Home phone number	[Phone | telephone| Phones| home |tel. | tel | ph]
Mother's maiden name	[mothers maiden name |mother's maiden name | mother maiden name | mmn]
Credit card number	[credit card number| creditno| creditnum| ccnum| ccno| cc#| card number| amex| master card]
Credit card expiry date	[credit expire date |expire date| e-date| expdate| expiration date]
Credit card type	[credit card type| card type| Ctype]
Credit card security number	[security number| verification number]

Through manual search using Google, different synonyms corresponding to the above identity information were learned and were used in implementing our identity information search module.

Table 1 depicts the selected keywords. All these keywords will be referred to, in the rest of the paper, as the identity search keywords.

Figure 2 depicts our identity information detection (IID) algorithm. The searching mechanism of the IID algorithm has three phases. The first phase is described in steps 1 through 8. During this phase, first of all the search engine's API is invoked to find web pages containing the name of the target user and at least one of the identity search keywords. In steps 6 and 8, the content of each document returned from the search is converted into text and saved in a text document. This is because the returned documents could be in different formats (e.g., PDF, DOC, XLS, HTML) and converting them into text enables us to easily parse them.

The second phase is described in steps 9 to 11. In this phase, a function named *KeywordFilter* described in Figure 3 is invoked in order to identify the exact list of identity search keywords appearing in each document. The keyword filter is designed while taking into consideration possible synonyms for each of the identity keywords described earlier.

The third phase covers steps 12 to 17. In this phase a (keyword) value filter function named *ValueFilter* described in Figure 4 is used to identify the exact list of values corresponding to the identity search keywords appearing in each document. The value filter is designed while taking into consideration all the different formats used to express the identity keyword values.

Finally, document names along with a list of identity keywords and values are displayed for only those documents that at least have one value corresponding to an identity keyword. The rest of the documents are discarded.

The prototype of our retrieval engine was implemented and tested using Google API. Figure 5 represents a screen shot of one of the files created by our tool as a result of searching for an individual named John Q. Jones. Each block of information corresponds to the identity information found in a single document for this particular individual. Each block of information corresponds to an identity information pattern, which as mentioned earlier will be processed (by the fraud detector) to detect possible inconsistencies with submitted applications by John Q. Jones.

```
Algorithm: keywordFilter(d)

Input: A document d.
Output: a list Keys of keywords found in the given document.
Uses: Set of regular expressions that returns a keyword if found or ∅ otherwise, and a list of keywords ArrayofK that we are going to search for in the document.

Steps:

 let ArrayofKeys = ∅, a placeholder for the list of keys
  found in the document

  for each k_i ∈ ArrayofK do:
      KeyFound=KeyRegularExpression(d, k_i)
      if KeyFound ≠ ∅
          ArrayofKeys [i]=k_i
      endif
 endfor
return ArrayofKeys
```

Figure 3. Keyword filtering algorithm that uses a set of regular expressions to identify the identity keywords appearing in the document

5. Identity Fraud Detector

We illustrate in this section our proposed identity fraud detection strategy and algorithms.

Our identity fraud detection approach consists of two major phases: the derivation of individual profiles and the analysis of the derived profiles using rule-based decision tree. We illustrate each of the two phases in the following.

5.1 Shared identity information

Let P be the set of identity patterns returned by a targeted search. Each identity pattern $pi \in P$ is represented by a k-dimensional attribute vector $<p_{i1},\ldots, p_{ik}>$. Possible examples of attributes include the following:

<Social security number, Date of birth, Address, Telephone number, Mother maiden name, Credit card number, Credit card expiry date, Credit card type, Credit card security number>

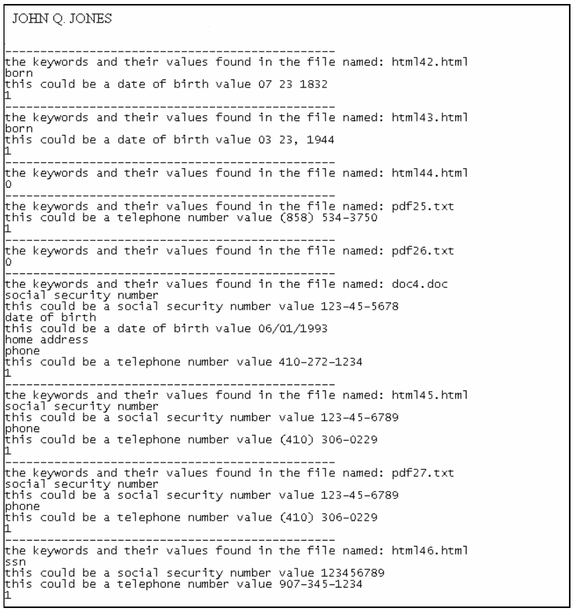
We exclude the name simply because at this stage we are dealing with identity information belonging to homonyms. A typical identity pattern returned from a targeted search might include only a subset of the k attributes set. The missing (or undefined) fields will simply be considered non-applicable (NA), and will not be used for the matching.

```
Algorithm: ValueFilter(d, ArrayofKeys)

Input: A document d, along with the list of keywords
ArrayofKeys found in that document.

Output: An array of m rows and 2 columns, each row has one
of the keywords found in the document along with its value,
or an empty array otherwise if no keyword-values were found.

Uses: Set of regular expressions that returns a value of a
keyword if found or ∅ otherwise.

Steps:
 let ArrayofValues = ∅, a list that will contain the
 keywords found in the document along with their
 values.

   for each k_i ∈ ArrayofKeys do:
      ValueFound=ValueRegularExpression(d, k_i)
       if ValueFound ≠ ∅
            ArrayofValues [i][1]= k_i
            ArrayofValues [i][2]= ValueFound
       endif
  endfor
 return ArrayofValues
```

Figure 4. Value Filtering algorithm that uses a set of regular expressions to identify the values for the identity keywords appearing in the document

Figure 6 illustrates sample identity patterns. In this example the following five identity attributes are considered: social security number (ssn), date of birth (dob), mother maiden name (mmn), address (addr) and phone number (tel).

The retrieval engine uses a name and a set of keywords to search the identity information source (e.g., the Web), and returns a collection of identity information related to the identity claim corresponding to the application being checked. As a result the returned identity information may correspond to several different individuals, all having the same name. For instance, we can have in the search result several social security numbers corresponding to the same name, which necessarily means that different individuals actually share the searched name. But this could also mean that the same person is impersonating all these different individuals by simply changing some of the identifying attributes. For the purpose of fraud detection, we need to identify and isolate the trail of identity patterns that relate to the individual submitting the application. We determine the patterns related to the applicant by determining direct and indirect connections between the application pattern and the patterns retrieved from the identity information source.



```
JOHN Q. JONES

-----------------------------------------------
the keywords and their values found in the file named: html42.html
born
this could be a date of birth value 07 23 1832
1
-----------------------------------------------
the keywords and their values found in the file named: html43.html
born
this could be a date of birth value 03 23, 1944
1
-----------------------------------------------
the keywords and their values found in the file named: html44.html
0
-----------------------------------------------
the keywords and their values found in the file named: pdf25.txt
this could be a telephone number value (858) 534-3750
1
-----------------------------------------------
the keywords and their values found in the file named: pdf26.txt
0
-----------------------------------------------
the keywords and their values found in the file named: doc4.doc
social security number
this could be a social security number value 123-45-5678
date of birth
this could be a date of birth value 06/01/1993
home address
phone
this could be a telephone number value 410-272-1234
1
-----------------------------------------------
the keywords and their values found in the file named: html45.html
social security number
this could be a social security number value 123-45-6789
phone
this could be a telephone number value (410) 306-0229
1
-----------------------------------------------
the keywords and their values found in the file named: pdf27.txt
social security number
this could be a social security number value 123-45-6789
phone
this could be a telephone number value (410) 306-0229
1
-----------------------------------------------
the keywords and their values found in the file named: html46.html
ssn
this could be a social security number value 123456789
this could be a telephone number value 907-345-1234
1
```

Figure 5. Sample search results returned by our tool for an individual named John Q. Jones. Each block of information represents an identity information pattern in our framework

Two patterns are directly connected if they share at least one attribute value. Two patterns p and q have an indirect connection, if there is a sequence of directly connected patterns linking them. In other words, there exists a sequence of patterns *p1… pn*, such that (p, p_1), (p_n,q), and (p_i, p_{i+1}), $\forall\ i \in \{1,\ldots, n-1\}$, are each a direct connection.

For the purpose of fraud detection, we trim the returned set P of identity patterns, and retain only the patterns having direct or indirect connections with the application pattern; let P^+ denote the subset of P containing all such patterns.

Let p_0 denote the application pattern and let G denote a set of patterns such that $G = P^{+} \cup \{p_0\}$.

As an example, let's consider the sample patterns depicted by Figure 6. Figure 7 depicts a tree structure exposing the direct and indirect connections between the sample patterns shown in Figure 7. We assume in this example that p_0 is the application pattern and the set of retrieved patterns is P, $P= \{p_1, p_2, p_3, p_4\}$. The set of connected patterns is G, $G= \{p_0, p_1, p_2, p_3\}$. For instance, patterns p_0 and p_2 share two attributes, namely *ssn* and *mmn*. Pattern p_4 has no connection (direct or indirect) with p_0, so it is excluded from the fraud analysis. It is assumed in this case that p_4 does not belong to the applicant but belongs to another person that shares his name.

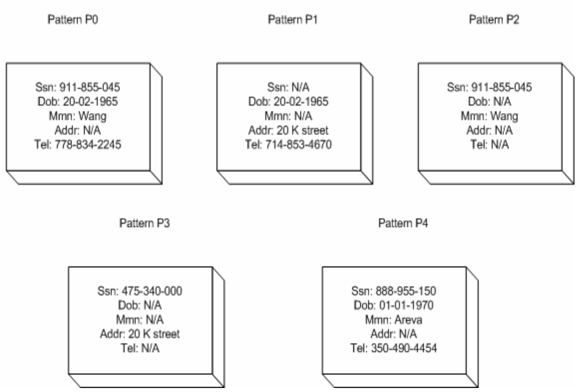


Figure 6. Examples of identity patterns based on 5 attributes

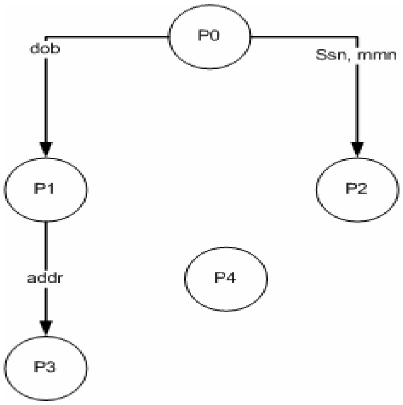


Figure 7. Direct and indirect connections between identity patterns

For the purpose of fraud detection, we analyze the link between every pattern-pair from G which are directly connected. Specifically we convert every pattern-pair that are directly connected into a single feature vector characterizing the underlying relationship. For each pattern-pair $< p_i, p_j > \in G \times G$, we derive a feature vector $v_{ij} = [\delta_{ijl}]_{1 \le l \le k}$, such that:

$$\delta_{ijl} = \begin{cases} ? \;\; if \;\; ((p_{il} = na) \;\; or \;\; (p_{il} = na)) \\ 1 \;\; if \;\; p_{il} = p_{jl} \\ 0 \;\; otherwise \end{cases} \qquad (1)$$

Where "?" denote a missing value, which may be either 1 or 0, but this is unknown.

The derived feature vector captures numerically the existence (or lack) of shared identity information in the corresponding pattern-pair. Let V denote the set of all the feature vectors derived from G. The derived set of feature vectors corresponding to the previous example (see Figure 7) is as follows:

$$V = \left(v_{01} = \begin{vmatrix} ? \\ 1 \\ ? \\ ? \\ 0 \end{vmatrix}, v_{02} = \begin{vmatrix} 1 \\ ? \\ 1 \\ ? \\ ? \end{vmatrix}, v_{03} = \begin{vmatrix} 0 \\ ? \\ ? \\ 0 \\ ? \end{vmatrix}, v_{12} = \begin{vmatrix} ? \\ ? \\ ? \\ ? \\ ? \end{vmatrix}, v_{13} = \begin{vmatrix} ? \\ ? \\ ? \\ 1 \\ ? \end{vmatrix}, v_{23} = \begin{vmatrix} 0 \\ ? \\ ? \\ ? \\ ? \end{vmatrix} \right)$$

5.2 Fraud detection

We present, in this section, our fraud detection approach, followed by an overview of the RBDT-1 method, and the initial rule base.

5.2.1 Approach

Our fraud detection approach consists of analyzing the coherence or consistency of the shared identity information between patterns. For instance, two identity certificates attributed to the same individual are expected to bear the same birth date and mother maiden name, when such information is available. The feature vectors describing patterns connections are used as basis for such analysis. As an outcome of the analysis, patterns connections are classified in one of four categories: *normal, suspicious-low, suspicious-high*, or *fraudulent* represented by the labels *N*, *S-*, *S+*, and *F*, respectively. A normal connection is a connection for which no anomaly has been found. We consider two strains or levels of suspicion, *suspicious-low* and *suspicious-high* that refer to unlikely and highly unlikely situations, respectively. A suspicious connection may potentially involve some fraud; the judgment is not definitive, and as such it calls for further inspection. In contrast, a fraudulent connection is a profile involving definitely some fraud; the judgment here is final.

Our fraud detector is implemented as a decision tree that takes pattern-pair feature vectors as an input, and provides as an output a fraud decision. The tree is built and updated dynamically based on a rule base which encodes expert knowledge or common sense understanding of the notion of fraudulent or inconsistent behaviour. No previous fraud instances should be required. The size of the decision tree resulting from such process is expected to be manageable because typically application forms involve sparse and limited identity attributes.

We use a new rule-based decision tree method named RBDT-1 [5, 6] to transform the set of rules into a decision tree. In the rest of this subsection, we give an overview of the RBDT-1 method, and then summarize and discuss a sample of the rules.

5.2.2 The RBDT-1 method

RBDT-1 is a new rule based decision tree method for learning a decision tree from a set of decision rules that cover some data instances. *RBDT-1* method uses a set of declarative rules as an input for generating a decision tree. The method's goal is to create on-demand a short and accurate decision tree from a stable or dynamically changing set of rules. The rules used by *RBDT-1* could be generated either by an expert, or by an inductive rule learning program. There is a major difference between building a decision tree from examples and building it from rules. When building a decision tree from rules the method assigns attributes to the nodes using criteria based on the properties of the attributes in the decision rules, rather than statistics regarding their coverage of the data examples. To derive the tree, the *RBDT-1* method uses in sequence three different criteria to determine the fit (best) attribute for each node of the tree, which are referred to as *the attribute effectiveness (AE)*, *the attribute autonomy (AA)*, and *the minimum value distribution (MVD)*.

- *Attribute Effectiveness*: the first criterion to be examined for the attributes prefers an attribute which has the most influence in determining the decision classes. In other words, it prefers the attribute that has the least number of "don't care" values for the class decisions in the rules, as this indicates its high relevance for discriminating among rule-sets of given decision classes.
- *Attribute Autonomy*: the second criterion to be examined for the attributes in the *RBDT-1 method*. This criterion is examined when the highest *AE* score is obtained by more than one attribute. This criterion prefers the attribute that will decrease the number of subsequent nodes required ahead in the branch before reaching a leaf node. Thus, it selects the attribute that is less dependent on the other attributes in deciding on the decision classes.
- *Minimum Value Distribution (MVD)*: is concerned with the number of values that an attribute has in the current rules. When the highest *AA* score is obtained by more than one attribute, this criterion selects the attribute with the minimum number of values in the current rules. This criterion minimizes the size of the tree because the fewer the number of values the fewer the number of branches involved and consequently the smaller the tree will be.

In the decision tree building process, we select the fit attribute that will be assigned to each node from the current set of rules *CR* based on the attribute selection criteria outlined above. *CR* is a subset of the decision rules that satisfy the combination of attribute values assigned to the path from the root to the current node. *CR* will correspond to the whole set of rules at the root node. From each node a number of branches are pulled out according to the total number of values available for the corresponding attribute in *CR*. Each branch is associated with a reduced set of rules *RR* which is a subset of *CR* satisfying the value of the corresponding attribute.

If *RR* is empty, then a single node will be returned with the value of the most frequent class found in the whole set of rules. Otherwise, if all the rules in *RR* assigned to the branch belong to the same decision class, a leaf node will be created and assigned a value of that decision class. The process continues until each branch from the root node is terminated with a leaf node and no more further branching is required.

5.2.3 Rule base

Based on the five attributes patterns considered in our previous example, we derived 82 rules for the initial implementation of our proof concept. Table 2 illustrates a decision table for a sample of the rules. In the rules "yes" corresponds to $\delta_{ijl} = 1$, "no" corresponds to $\delta_{ijl} = 0$, "na" corresponds to $\delta_{ijl} = ?$ and * is a placeholder for all possible values of δ_{ijl} (i.e. 1, 0, "?").

Table 2: Sample of the rules in the rule-base

Attributes **Rules #**	**ssn**	**mmn**	**dob**	**add**	**tel**	**decicison**
1	no	yes	yes	na	na	s-
2	no	yes	yes	na	no	s-
3	na	no	yes	yes	*	s-
4	no	*	yes	yes	*	s+
5	yes	*	no	*	*	f
6	yes	no	yes	*	*	f
7	yes	no	na	*	*	f
8	*	*	*	no	yes	f
9	no	*	*	na	yes	f
10	no	no	*	no	no	n
11	na	no	*	no	no	n
12	no	na	*	no	no	n
13	na	na	*	no	no	n
14	no	no	*	na	no	n
15	yes	yes	yes	yes	yes	n

While some of the rules are straightforward, many require some explanations. A straightforward fraud case is when the social security numbers match and either the dates of birth or the mother maiden names do not such as rules #5, #6 and #7 in Table 2. Some other straightforward cases of fraud are when the home telephone numbers match while the addresses do not such as rule #8.

In both cases one could assume that the same individual is impersonating two different individuals: in the first case by changing either the mother maiden name or the date of birth, and in the second case by using different locations.

Two straightforward normal cases are when either none of the attributes match or all the attributes match such as rules #10 and #15, respectively. We assume in the case when none of the attributes matches that the non-matching patterns belong to different individuals and have not been used by the applicant (in some past attempt to defraud the system). Obviously, the case when all the attributes match is regular. A variant of this case is when everything else match except the telephone numbers; this is considered normal because it could simply be the case that the same individual owns several telephone lines.

According to the degree of rarity the case is classified as either highly suspicious or lightly suspicious. For instance, considering that we are dealing with homonyms, a case where the social security numbers do not match, while the addresses, dates of birth, and mother maiden names match, such as rule #4, is extremely unusual. Because this simply means that there are two different individuals (different ssn), who are homonyms, born the same day, from homonym mothers and living at the same location. This is highly suspicious and requires further verification. If these two individuals (born on the same day with the same mother maiden name) were living in different locations, with different phone numbers, then the situation could be considered as a coincidence, although still uncommon. We classify such occurrence as lightly suspicious. Further investigation would also be required in this case as well, but such investigation does not have to be as thorough as in a highly suspicious case.

The decision tree obtained, by applying the RBDT-1 method to the rule base, is illustrated in Figure 8. The resulting tree consists of 57 rules; the distribution of the different decisions is summarized in Table 3.

Table 3: Rules distribution summary for the RBDT-1 decision tree

Decision	#Rules
Normal (N)	39
Fraud (F)	10
Suspicious-Low (S-)	7
Suspicious-High (S+)	1
Total	57

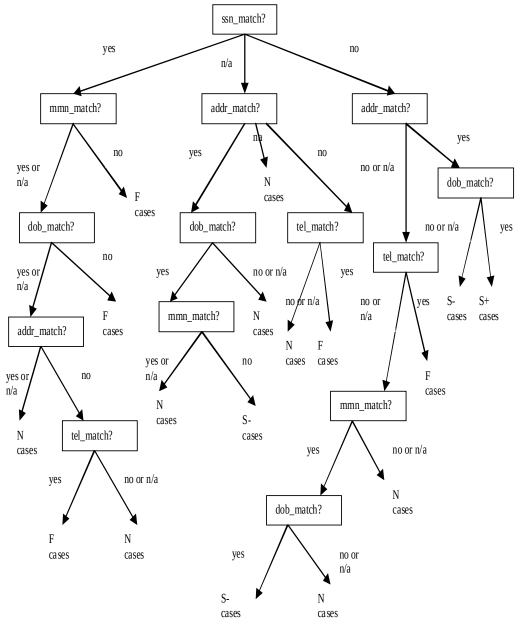


Figure 8. The decision tree produced by RBDT-1.

6. Evaluation

We present, in this section, the evaluation of the different components of our fraud detection framework. We start by presenting the evaluation method and data collection, and then we present and discuss the evaluation results obtained for the identity information retrieval engine, followed by the fraud detector.

6.1 Evaluation method and data

The most common way of evaluating frameworks like the one proposed in this paper consists of finding and using some public dataset containing real identity fraud information. As expected, however, the evaluation of our framework faces difficulty in obtaining real data to effectively analyze and test our system. Usually data containing real identity (fraud) information is restricted from being used due to privacy issues. An alternative approach may consist of generating synthetic fraud data. In this work, we collected real identity information online and then generated and injected synthetic fraud information in the data. Having, however, a dataset is not enough to carry out the evaluation. We need an unbiased mechanism to label the data. In the absence of a domain expert to label the data, we have decided to adopt a comparative labelling mechanism in this work. More specifically, we used the communal scoring approach by Phua *et al.* described above in the related work to produce an initial set of labels, and then we compared the produced labels to the labels obtained while using our proposed approach. Using such comparative evaluation allows us to assess the relative strength of our approach.

To obtain real identity information, we conducted some white hat Google hacking experiments over two weeks. During this study, we searched manually for identity information using Google. We used different words that express identity information such as the social security number and searched in different types of documents, and were able to collect sensitive identity information for living as well as for dead persons. The study allowed us to establish a database of online identity information related to 154 different individuals, which serve as basis for the validation of our identity fraud detection framework. Details of the study and corresponding results can be found in [14].

6.2 Evaluation of the information retrieval engine

In order to evaluate our search algorithms, described earlier, firstly, we used Google search engine to manually search for pages containing identity information for only 70 different names out of the 154 names identified in our initial manual search described in the previous section. Those 70 names were the subset of the 154 names for which identity information were still available online at the time of this evaluation; the information corresponding to the remaining names were removed between the time of the initial data collection and the online evaluation reported here. In our search query we used the search identity keywords listed in Table 1 along with each of the 70 names.

The search results for each name were classified into two categories:

- *Keyword-only documents*, which are documents that contain some keywords but no values for any of them.

- *Keyword-value documents,* which are documents that contain at least one value corresponding to an identity keyword.

Secondly, we ran our identity information retrieval tool for each of the 70 names used in the previous manual search. The results of each run consist of document names along with a list of identity keywords and values displayed only for those documents that contain at least one value corresponding to an identity keyword. The numbers of such documents, which are categorized as Keyword_Value documents, as well as the execution time that each search takes are also displayed in the output.

These experiments were conducted on a COMPAQ PC (2 Ghz, 512 Mb of RAM) with AMD Sempron 3000+ and the programs were written using PYTHON language.

We used three metrics to assess the performance of our automated information retrieval scheme, namely recall, precision and false alarm. In the context of our identity information retrieval scheme, these metrics can be defined as follows:

Recall: is the proportion of the correct Keyword_Value documents retrieved by the application from the set of total Keyword_Value documents calculated in the manual search. Recall can be calculated as in (2):

$$\mathrm{Recall} = \frac{\text{Proportion of Correct Keyword\_Value documents calculated by the tool(CKVT)}}{\text{Total number of Keyword\_Value documents calculated manually(KVM)}} \quad (2)$$

Precision: is the proportion of the correct Keyword_Value documents retrieved by the application from the set of total Keyword_Value documents claimed in the application output. Precision can be calculated as in (3):

$$\mathit{Precision} = \frac{\text{Proportion of Correct Keyword\_Value documents calculated by the tool(CKVT)}}{\text{Total number of Keyword\_Value documents calculated by the tool(KVT)}} \quad (3)$$

False Alarm: is the proportion of the false Keyword_Value documents retrieved by the Identity Information Detector (IID) from the set of total Keyword_Only documents calculated in the manual search. False alarm can be computed as in (4):

$$\mathit{False\ Alarm} = \frac{\text{Proportion of False Keyword\_Value documents calculated by the Tool (FKVT)}}{\text{Total number of keyword\_Only documents calculated Manually (KOM)}} \quad (4)$$

We calculated the recall, precision, and false alarm rates for each of the 70 names, based on corresponding search results. Overall we obtained an average recall of 97.16%, an average precision of 81.82 % and an average false alarm of 13.33%. In general these figures are acceptable since we are collecting information from documents with different formats and structures. For instance, comparable results were obtained in [11] and [12] although the information in their case was collected only from one source of documents which are curriculum vitas.

6.3 Evaluation of the fraud detector

We present, in this section, the application fraud data used to validate our fraud detector and the technique used to label the data. Finally we present the results produced by our fraud detector based on the labelled data.

6.3.1 Application fraud data

In order to evaluate our application fraud detector, we need instances of labelled data both with normal, fraud and suspicious cases and then we need to label the data with our proposed fraud detector and compare both labels.

As mentioned above, due to privacy and confidentiality reasons, obtaining real identity fraud information is extremely difficult. Usually the available real data is encrypted and key identity attributes are removed which makes it ineffective for evaluating our detector.

To overcome this issue we assume that the data that we collected in our white hat Google hacking experiments are application forms submitted by applicants applying for an identity certificate, which yields 154 normal applications. In addition we used the names of each of the 154 applicants to search for identity patterns containing identity information that share the same name.

Since the data that we collected does not include fraud instances, we created some synthetic fraud data based on obvious fraud concepts. Specifically, the main obvious fraud concept used in our evaluation was based on the case where pair of applications bear the same social security number and have either different dates of birth or different mother maiden names. Based on this fraud concept, we created synthetic data based on all the possible value combinations of the identity attributes involved. The synthetic data produced is (2×3^3) – 9(redundant cases) = 45 fraud data instances. So, overall our evaluation dataset involved 199 data instances consisting of 154 normal cases and 45 fraud cases.

6.3.2 Data labeling

We used a methodology based on a technique proposed by Phua *et al.* in [8] for labelling the data instances described in the previous section as normal, fraud or suspicious.

In order to make the paper self-contained, we briefly describe Phua *et al.* technique which we will refer to as CASS for communal analysis suspicion scoring. CASS is a technique for generating numeric suspicion scores for credit applications based on implicit links to other previous applications over both time and space.

Every new application v_i is pair-wise matched against previous scored applications within a window W .

The attribute vector y_{ij} between v_i and v_j which captures the relationship for an application-pair is defined as:

$y_{ij} = \{y_1{}^{ij}, y_2{}^{ij}, ..., y_N{}^{ij}\}, \forall i, j$

Where N is the total number of attributes and $y_k{}^{ij}$ is the individual match score of each pair-wise attribute: $y_k{}^{ij} = y_k\left[a_{ik}, a_{jk}\right]$, $1 \le k \le N$.

The match score of each pair-wise attribute y_k could be Boolean 1 for a match or 0 for a non match. In this case the maximum possible score of an application–pair is $\sum_{k=1}^{N} y_k{}^{ij} = N$. By assigning weights to the attributes, the maximum possible score of an application–pair would be $\sum_{k=1}^{N} y_k^{ij} = 1$.

With their technique, Phua *et al.* label a new application v_i by first linking it with an existing application v_j from either a black list or a white list or as anomalous, or by considering it unlinked. Linkage between v_i and v_j is denoted as $v_i \xrightarrow{L} v_j$, where L is a label, corresponding to *fraud*, or *normal* or *anomalous*. After establishing the linkage, the communal suspicion score $W_{communal_{ij}}$ for the linked application-pair is computed accordingly.

Phua *et al.* assume that the black list is made up of actual identity applications previously found to be fraudulent. As a result, any new application that contains similar identity information to those applications in the black list is considered as a fraud.

For the black list, Phua *et al.* define the communal link derived from pair-wise matching of attributes $W_{communal_{ij}}$ as in (5):

$$W_{communal_{ij}} = \begin{cases} 1 \text{ and } v_i \xrightarrow{fraud} v_j \text{ if } v_j \text{ is a known fraud} \\ \qquad\qquad\qquad \text{and } \sum_{k=1}^{N} S(y_k^{ij}) \ge T_{fraud} \\ 0 \quad \text{otherwise} \end{cases} \tag{5}$$

Where T_{fraud} is an integer threshold for linking a new application to the black list ($0 \le T_{fraud} \le N$) and S is a string similarity metric, which can be exact matching or fuzzy matching. We use in our evaluation exact matching.

Although the white list is the opposite of the black list, it is not made up necessarily of the actual identity applications that are not fraudulent. The white list consists of a set R of N relationships (or rules) defined as normal: $R = [R_1, R_2, R_3, ..., R_N]$. A weight WR_k is associated with each relationship R_k $(1 \le k \le N)$, where $0.5 \le WR_k \le 1$. The set of weights $W_{normal} = [WR_1, WR_2, ..., WR_N]$ is sorted in ascending order while the set of relationships R is ranked in descending order of WR_k.

For the white list, the communal score $W_{communal}$ is defined as in (6):

$$W_{communal_{ij}} = \begin{cases} [WR_x * \sum_{k=1}^{N} S(y_k^{ij})] \text{ and } v_i \xrightarrow{normal} v_j \text{ if } y_{ij} \in R \\ \qquad\qquad\qquad \text{and } \sum_{k=1}^{N} S(y_k^{ij}) \ge T_{normal} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{6}$$

Where T_{normal} is an integer threshold for linking a new application to the white list ($0 \le T_{normal} \le N$), and WR_x is the weight associated with relationship $R_x \in R$ matching with y_{ij}.

Phua et al. assume that anomalous application pairs correspond to linked applications that are in neither the black list nor the white list. The communal score $W_{communal}$ is defined in this case as in (7):

$$W_{communal_{ij}} = \begin{cases} \sum_{k=1}^{N} S(y_k^{ij}) \text{ and } v_i \xrightarrow{anomalous} v_j \text{ if } y \notin R \\ \qquad\qquad\qquad \text{and } \sum_{k=1}^{N} S(y_k^{ij}) \ge T_{normal} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{7}$$

Unlinked application-pairs are the results of too few attributes (below the selected thresholds) or no attribute match at all. Phua *et al.* state that there is a strong possibility that many fraudulent applications that share few (below the fraud threshold) identity attributes or no attribute with those in the black list would not be detected and would be accepted as normal.

In order to make a decision for a new application, the total suspicion score for that application is calculated based on all the linked application pairs' scores, and then a final classification as fraud, normal or anomalous is made [8].

In the process of labeling our data using Phua *et al.*'s technique, we populated the black list with identity information of four synthetic fraud applications crafted based on the obvious fraud case explained in the previous section. These will be used to match against our 45 synthetic fraud applications mentioned in the previous section. We populated the white list with 15 normal relationships defined by Phua *et al.* in [15] after adapting them to fit the five attributes used in our fraud detector. We randomly selected 49 patterns from the identity patterns that we collected online and labeled them as normal to link them to the white list.

We assumed that the suspicion scores of the applications linked to the black list were very high, and those linked to the white list were very low. Thus, in this case we labeled a new application linked to an application in the black list with $W_{communal} = 1$ as a fraud application. We labeled the applications linked to the white list with $W_{communal} > 0$ and the unlinked applications as normal applications and labeled those linked as anomalous with $W_{communal} > 0$ as suspicious.

6.3.3 Evaluation

To evaluate our application fraud detector we removed the labels of the data and allowed our system to produce its own labels, and then compared them to the data labels produced by CASS. The outcome of the comparison is the *Match Rate (MR)*, when the labels produced by both techniques coincide. The *Non Match Rate (NMR)*, which is the complement of the MR (NMR=1-MR) measures the disagreement between both techniques. Note that a high NMR doesn't necessarily mean a weakness of either technique. A closer error analysis needs to be done to find out which technique is actually at fault.

We chose the values 2 and 3 for the normal and fraud thresholds, which are considered intermediate values since the application-pair matching score is based on five attributes. As a result of setting the thresholds for fraud and normal to a combination of the above two values (2 and 3) we produced four different sets of labels for our data. We compared the labels produced by our proposed fraud detector to each of the four sets of labels and computed the match rate by our method which is summarized in Table 4.

Table 4: Summary of the match rates produced by our fraud detector for different thresholds values when compared to CASS-labeled data

T_{fraud}	T_{normal}	**Match Rate**
2	2	87%
3	3	82%
3	2	77%
2	3	92%

Based on the selected thresholds, the best match rate produced by our proposed fraud detector was 92% for the data labelled by CASS using a combination of $T_{normal} = 3$ for the white list and the anomalous and $T_{fraud} = 2$ for the black list.

To analyze this result, we have to discuss the labels produced by the CASS approach. By reviewing the 45 synthetic fraud applications that we created, we discovered that 12 fraud applications were labelled as unlinked by CASS, and as a result were considered as normal and got accepted. These 12 applications are cases where two applications share only one attribute value (below the black list threshold) that happens to be the social security number. Although these applications share only one attribute, they correspond to an obvious fraud case because the social security number is a unique attribute and hence requires the rest of the attributes to be identical. Thus, in that sense our application fraud detector produces a correct fraud label for all the 45 fraud applications which, however, does not match the CASS labels. CASS produces the wrong labels because, as explained above, the method depends on a threshold which is not effective in detecting such fraud cases.

For the 154 normal applications, while our application fraud detector labels them as normal, CASS labelled 3 of the 154 applications as anomalous and the rest as normal. Examination of these 3 CASS-labelled anomalous applications shows that while the threshold for a linkage between each of the applications and a normal application is met there is no corresponding matching relationship in R. This could possibly be addressed by expanding the initial set of normal relationships used for the evaluation.

Overall our proposed system performs well when assessed against data labelled using CASS, which underscores its potential as a strong fraud detector.

7. Conclusion

In the last several years, identity theft has been on the rise. Unfortunately, the Internet has been facilitating this phenomenon since it represents a tremendous and open repository for sensitive identity information available for those who know how to find them, including fraudsters. We have presented in this paper an unsupervised application fraud detection framework that analyzes online identity patterns using an intelligent decision model to identify fraudulent applications. We designed a web-based identity information retrieval engine capable of finding and collecting online identity information. We defined heuristic rules for detecting fraudulent applications and used a rule-based decision tree method for transforming the rules into a decision tree. To evaluate our fraud detector we used a mix of real data collected online and synthetic data to induce some fraud cases. The data was treated as identity applications and labelled using a technique called CASS as normal, fraud or suspicious. The data was labelled four times based on the CASS approach by using different combinations of normal and fraud thresholds and compared to the labels produced by our fraud detector. The best match rate produced by our detector was 92%. The non-matching cases happened to be due to labelling errors by the CASS approach. Overall our approach shows strong promise for online identity application fraud detection.

In future work, we intend to extend and apply our fraud detection approach to other possibly more traditional data sources.

References

[1] Crown "Identity fraud: A study", Economic and Domestic Secretariat Cabinet Office, www.homeoffice.gov.uk/docs/id_fraud-report.PDF, 2002.

[2] W. Wang; Y. Yuan; N. Archer. "A Contextual Framework for Combating Identity Theft", Security & Privacy Magazine, IEEE Volume 4, pp. 30 – 38, 2006.

[3] Bolton, R. J. and Hand, D. J. "Unsupervised profiling methods for fraud detection", Credit Scoring and Credit Control, Vol. 2, pp.5–7, 2001.

[4] I. F. Imam, and R. S. Michalski "Should Decision Trees be Learned from Examples of From Decision Rules?", Source Lecture Notes in Computer Science. In Proceedings of the 7th International Symposium on Methodologies, 689, pp. 395–404, 1993.

[5] A. Abdelhalim, I.Traore.'Converting Declarative Rules into Decision Trees', International Conference on Computer Science and Applications, San Francisco, USA, 20-22 October, 2009.

[6] A. Abdelhalim, I.Traore, B. Sayed. 'RBDT-1: a New Rule-based Decision Tree Generation Technique', 3rd International Symposium on Rules, Applications and Interoperability, Las Vegas, Nevada, USA: Nov 5-7, 2009.

[7] P. Burns, A. Stanley. "Fraud Management in the Credit Card Industry", Payment Cards Center Discussion Paper number 02-05, Federal Reserve Bank of Philadelphia.http://www.phil.frb.org/pcc/papers/2002/FraudManagement_042002.pdf, 2002.

[8] R. Wheeler, S. Aitken. "Multiple Algorithms for Fraud Detection", Knowledge-Based Systems, Vol. 13, No. 3, pp.93–99, 2000.

[9] C. Phua, V. Lee, K. Smith, R. Gayler, "A Comprehensive Survey of Data Mining-Based Fraud Detection Research", *Artificial Intelligence Review, submitted*, 2005. Available at: http://www.bsys.monash.edu.au/people/cphua/.

[10] P. A. Estevez, C. M. Held, C. A. Perez, "Subscription Fraud Prevention in Telecommunications Using Fuzzy

Rules And Neural Networks", Expert Systems with Applications, Vol. 31, pp. 337–344, 2006.

[11] L. Sweeney. "AI Technologies to Defeat Identity Theft Vulnerabilities", AAAI Spring Symposium on AI Technologies for Homeland Security, 2005.

[12] L. Sweeney. "Protecting Job Seekers from Identity Theft", IEEE Internet Computing, Vol. 10, No. 2, pp.74–78, 2006.

[13] Chan, P. K., Fan, W., Prodromidis, A. L. and Stolfo, S. J. "Distributed Data Mining in Credit Card Fraud Detection", Intelligent Systems, IEEE, Vol. 14, No. 6, pp.67–74, 1999.

[14] Abdelhalim, A. and Traore, I. "The Impact of Google Hacking on Identity and Application Fraud", Proc. IEEE Pacific Rim Conference, Communications, Computers and Signal Processing, pp. 240–244, 2007.

[15] Phua, C., Gayler, R., Lee, V. and Smith-Miles, K. "On the Communal Analysis Suspicion Scoring for Identity Crime in Streaming Credit Applications", European Journal of Operational Research, 195, pp. 595-612, 2009.

Amany Abdelhalim received an information system analyst degree from University of Helwan in 2000. She held a demonstrator position in information systems department from (2000- 2003). She received her MS.c in Computer Science from University of Helwan in 2003. She worked as an Assistant lecturer at the Department of Information Systems from (2003-2004), Helwan University, Cairo, Egypt. She is currently a PhD student in Electrical Computer Engineering Department at University of Victoria, Canada.

Issa Traore received an Aircraft Engineer Degree from Ecole de l'Air in Salon de Provence (France) in 1990, and successively two Master Degrees in Aeronautics and Space Techniques in 1994, and in Automatics and Computer Engineering in 1995 from *Ecole Nationale Superieure de l'Aeronautique et de l'Espace* (E.N.S.A.E), Toulouse, France. In 1998, Traore received a PhD in Software Engineering from Institute Nationale Polytechnique (INPT)-LAAS/CNRS, Toulouse, France. From June – October 1998, he held a post-doc position at LAAS-CNRS, Toulouse, France, and Research Associate (November 1998 – May 1999), and Senior Lecturer (June – October 1999) at the University of Oslo. Since November 1999, he has joined the Faculty of the Department of ECE, University of Victoria, Canada. He is currently an Associate Professor. His research interests include behavioral biometrics systems, intrusion detection systems, software security metrics, and software quality engineering. He is the founder and coordinator of the Information Security and Object Technology (ISOT) Lab (http://www.isot.ece.uvic.ca

Monitoring Data on Internet

Shishir Kumar, Amit Kumar

[1] Department of CSE,Jaypee Institute of Engineering & Technology,
Guna (MP) INDIA
shishir.kumar@jiet.ac.in

[2] Department of CSE,Jaypee Institute of Engineering & Technology,
Guna (MP) INDIA
amit.kumar@jiet.ac.in

Abstract: ***The problem of network intrusion has been tried to solve by many different approaches. In this rapidly growing technological world where system safety questioned by different network intrusions is growing per minute, network monitors help solve these problems of unwanted hacking and problems arising from the periodical failure in the working of wiring and protocols due to their complication. In our paper we have proposed an algorithm for monitoring data on Internet i.e. an algorithm for creating an essential network monitor. A network analyzer or, packet analyzer or sniffer, is computer software or computer hardware that can intercept and log traffic passing over a digital network or part of a network. As data streams flow across the network, the sniffer captures each packet and eventually decodes and analyzes its content. We have used this algorithm to create our own network monitor, which captures data transfers between the user system and any network resource. Aside from capturing and displaying the packets in a log table, it maintains a total traffic control by calculating the throughput of the network being monitored and the number of hosts, conversations, and protocols seen on the network. Filtering and searching options are also provided to the user for some parameters like protocols, source destination etc. This paper presents the details of the network packet analyser and about its working algorithm. It also shows its excellent performance.***

Keywords: Data Stream, Sniffer, Malware, Network Interface, Net-mask.

1. Introduction

Network monitors may be connected to the network to analyze the happening and attempt to identify the cause of the problem. Sniffers may also be used to identify the computers on the network that may cause problems such as using too much bandwidth, having the wrong network settings or running malware. Some hacking attempts can be found as they happen when network monitor shows sniffing of user's own servers for inappropriate traffic.

In this paper after detection of the network interfaces used by the system, the network interface is opened in a promiscuous mode. Packets are analyzed and their data is displayed to the user in hexadecimal format and character format [3]. This algorithm aims to act like an approximate network monitor in some contexts as it helps in the evaluation of the data flow on network.

WinPcap and jpcap network libraries have been used for implementation of proposed algorithm. WinPcap is the powerful and extensible architecture for low-level network analysis on Win32 platform. This architecture is the first open system for packet capture on Win32 and it fills an important gap between Unix and Windows. Packet filtering is made up of a kernel-mode component (to select packets) and a user-mode library (to deliver them to the applications). WinPcap includes an optimized kernel mode driver, called Netgroup Packet Filter (NPF), and a set of user-level libraries that are libpcap-compatible. Libpcap API compatibility was a primary objective in order to create across-platform set of functions for packet capture. WinPcap makes the porting of Unix applications to Win32 easier and it enables a large set of programs to be used on Win32 at once, just after a simple recompilation.

2. System Model

The system model of the proposed system is as shown in the above figure. In this moden the network adapter must be put into promiscuous mode in order to capture the traffic which can be unicast, multicast or, broadcast type, and is being sent to the machine running the sniffer software on wired broadcast and wireless LANs.

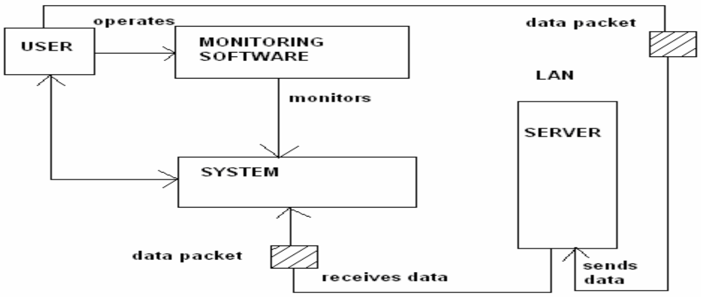


Figure 1. System Model

Promisc mode is a configuration of a network card that makes the network interface card pass all traffic it receives to the central processing unit rather than just packets addressed to it. Promiscuous mode[4] is often used to diagnose network connectivity issues. There are programs that make use of this feature to show the user all the data being transferred over the network. Some protocols like FTP and Telnet transfer data and passwords in clear text, without encryption, and network scanners can see this data.

3. Architecture to capture packets

The basic structure will have the modules of a filtering machine, two buffers (kernel and user) and a couple of libraries at user level. The filtering process is started by a user-level (`libpcap`-compatible) component that is able to accept a user-defined filter (i.e. “picks up all UDP packets”), compile them into a set of pseudo instruction (i.e. “if the packet is IP and the protocol type field is equal to 17, then return true”), send these instructions to the filtering machine, and activate that code. On the other hand, the kernel module must be able to execute these instructions; therefore it must have a “BPF virtual machine” that executes the pseudo-code on all the incoming packets. This kernel-side BPF-compatible filtering machine is a key point for obtaining good performance. WinPcap is made up of three modules, one at kernel level and two at user level; userland modules come under the form of Dynamic Link Libraries (DLLs).

First module is the kernel part (NPF) that filters the packets, delivers them untouched to user level and includes some OS-specific code (timestamp management) .Second module, *packet.dll*, is created to provide a common interface to the packet driver among the Win32 platforms. In fact, each Windows version offers different interfaces between kernel modules and user-level applications: *packet.dll* deals with these differences, offering a system-independent API. Programs based on *packet.dll* are able to capture packets on every Win32 platform without being recompiled. *Packet.dll* includes several additional functionalities. It performs a set of low level operations like obtaining the adapters’ names5 or the dynamic loading of the driver, and it makes available some system-specific information like the net-mask of the machine and some hardware counters (the number of collisions on Ethernet, etc.). Both this DLL and the NPF are OS-dependent and change between Windows 95/98 and NT/2000 because of the different OS architectures.

Second module, *WPcap.dll*, is not OS-dependent and it contains some other high-level functions such as filter generation and user-level buffering, plus advanced features such as statistics and packet injection. Therefore programmers can have access to two types of API: a set of raw functions, contained in *packet.dll*, which are directly mapped to kernel-level calls, and a set of higher level functions that are provided by *WPcap.dll* and that are more user-friendly and more powerful.

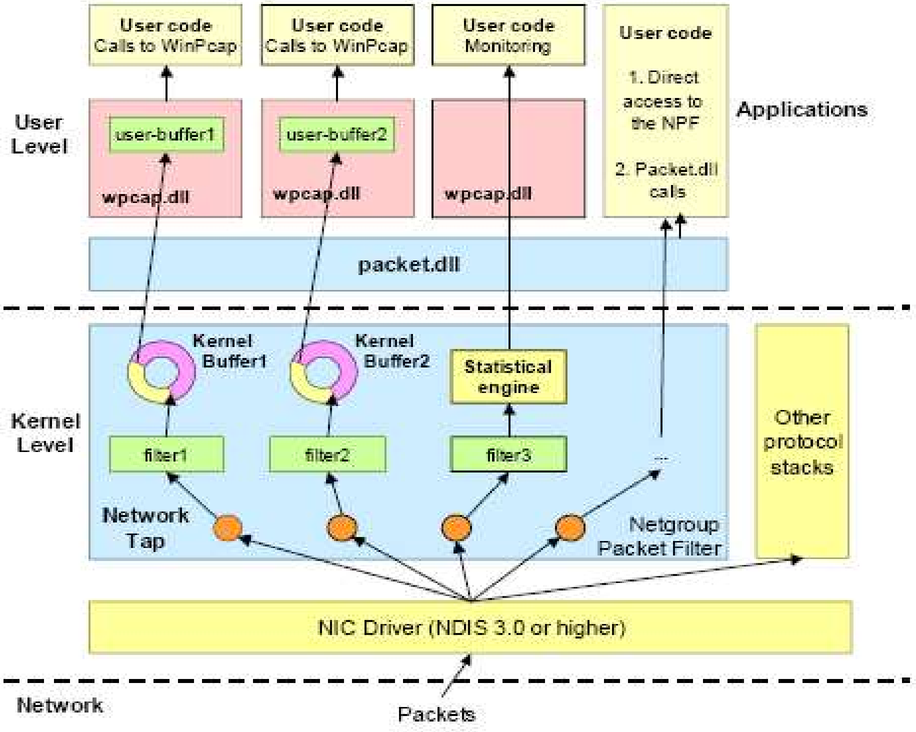


Figure 2. WinPcap Architecture

The latter DLL will call the former automatically; a single “high-level” call may be translated in several NPF system calls. Programmers will normally use *WPcap.dll*; direct access to *packet.dll* is required only in limited cases.

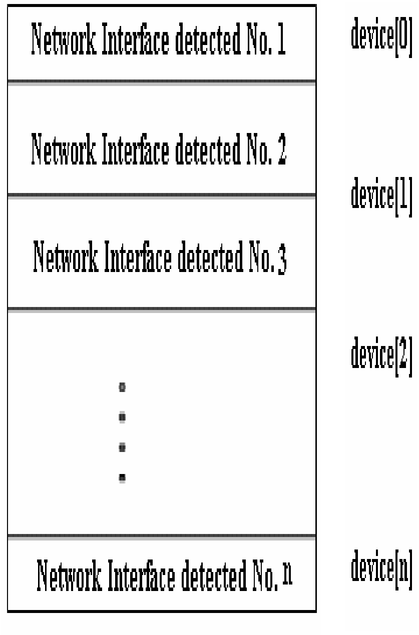

Network Interface detected No. 1
device[0]
Network Interface detected No. 2
device[1]
Network Interface detected No. 3
device[2]
Network Interface detected No. n
device[n]

Figure 3. Depicting the structure of variable ‘devices’, which is an object of Network Interface class.

4. Pseudo code of Implementation

Pseudocode for opening of network interfaces is as mentioned below:

```
NetworkInterface[] devices =
JpcapCaptor.getDeviceList();
for ( i = 1 to devices.length)
System.out.println(i+": "+devices[i].name
+ "("+ devices[i].description+")");
//print out its datalink name and
description
System.out.println(" datalink:
"+devices[i].datalink_name  + "(" +
devices[i].datalink_description+")");
//print out its MAC address
System.out.print(" MAC address:");
for (byte b : devices[i].mac_address)
System.out.print(Integer.toHexString(b&0x
ff) + ":");
System.out.println();
//print out its IP address, subnet mask
and broadcast address
for (NetworkInterfaceAddress a :
devices[i].addresses)
System.out.println(" address:"+a.address
+ " " + a.subnet + " "+ a.broadcast);
```

Pseudocode for capturing of packet in the network is as mentioned below:

```
public void receivePacket(Packet packet)
{
System.out.println(packet);
captor.processPacket(10,new
PacketPrinter());
captor.close();
```

4.1 Interface Detection

When you want to capture packets from the network, the first thing you have to do is to obtain the list of network interfaces on your machine. To do so, Jpcap provides JpcapCaptor.getDeviceList() method. It returns an array of Network Interface objects. A Network Interface object contains some information about the corresponding network interface, such as its name, description, IP and MAC addresses[5]’ and datatlink name and description.

Once you obtain the list of network interfaces and choose which network interface to capture packets from, you can open the interface by using JpcapCaptor.openDevice() method. We select the index number of an interface which we have to open then we have certain parameters on the basis of which we can make difference in the procedure. These parameters are as followed:

As all these parameters are used by JpcapCaptor.openDevice() function and it returns an instance of JpcapCaptor class. So further it can be used to capture the packets. Hence after following the above methods, functions, and parameter values only we can get the interface list at a particular node and can make it open for the interaction with internet so that packets[6] can enter and we can analyze them.

4.2 Representing a Packet

Once you obtain an instance of JpcapCaptor, you can capture packets from the interface. There are two major approaches to capture packets using a JpcapCaptor[4] instance: using a callback method, and capturing packets one-by-one.

In Jpcap, you can read the packets you saved using JpcapWriter by opening the file using JpcapCaptor.openFile() method. Similar to JpcapCaptor.openDevice()method, pcapCaptor.openFile() method also returns an instance of JpcapCaptor class. So you can use the same ways described in Capture packets from the network interface section to read packets from the file.

5. Performance Evaluation

In our paper we have discussed the case in which filtering is kept at a minimum level which gives the work an edge over other network packet analyzers in market on the parameter of speed .

In figure 4, and figure 5 depict the time lapses during catching of the packet as at that time the application gets involved in creating GUIs and doing network statistics on the captured network data.

In our algorithm, a network monitor is proposed where the earlier shown time lapses are kept almost negligible which results in the processing speed of capturing of packets being faster. Furthermore, WinPcap puts performance at the first place, thus it is able to support the most demanding applications In **Big O notation** the *O* stands for the "order" of the calculation. Here if a problem runs in $O(n^2)$ on one computer, then it will also run in $O(n^2)$ on all others, even though it may take longer or shorter depending on what resources they have. The exact amount of resources will depend on what machine or language is being used. This notation allows us to generalize away from the details of a particular computer. Performance of this algorithm is depicted via O(n) depicting that as the size 'n' of the input i.e. the network traffic to the program increases, then :

1. by the same 'n' factor the running time increases of the application as more processing steps would be needed to capture the data travelling in the network and also,

2. by the same 'n' factor the memory requirements increase as more memory space would be required to save the ' n ' size of transmitted data on the network.

Time Complexity = O(n)

Our algorithm is the most basic and viable algorithm for any network monitor. This algorithm is scaleable as it is suitably efficient and practical when applied to large situations i.e. a large input data set which here is heavy network traffic and handles them graceful manner. Here if the algorithm fails when the quantity of traffic increases then it is not scaleable which is not the case here.

In "scaling up"(scaling vertically) of our algorithm we add resources to a single node in a system i.e. involving the addition of CPUs or memory to a single user computer. In such a case the processing time of the algorithm would increase as system would be able to process the capturing functions much faster.

During scaling horizontally (or scale out) we add more nodes to the user system, such as adding a new computer to a distributed software application, and in such cases the processing time of the system decreases as the same number of steps would be performed by the system in a greater time period than originally required before scaling horizontally.

Table 1: Purpose of every Name procedure

Name	**Purpose:**
NetworkInterface intrface	Network interface that you want to open.
int snaplen	Max number of bytes to capture at once
boolean promics	True if you want to open the interface in promiscuous mode, and otherwise false. In promiscuous mode, you can capture packets every packet from the wire, i.e., even if its source or destination MAC address is not same as the MAC address of the interface you are opening. In non-promiscuous mode, you can only capture packets send and received by your host.
int to_ms	Set a capture timeout value in milliseconds.

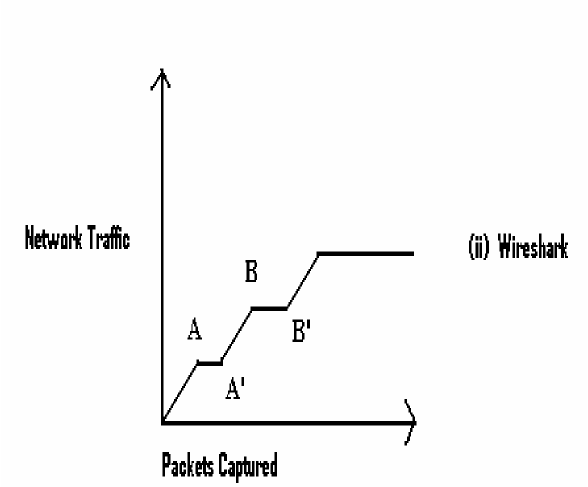


Figure 4. Packet Captured by Wireshark

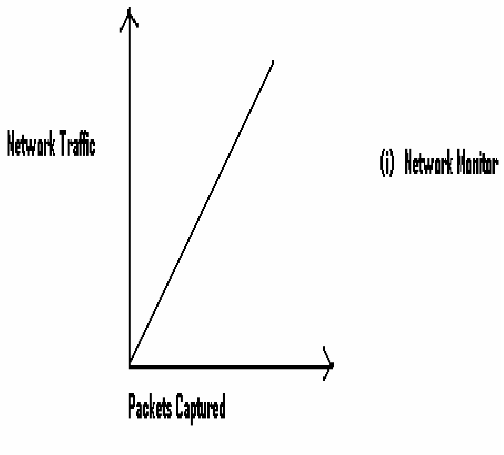


Figure 5. Packet Captured by Network Monitor.

6. WinPcap Performance

Packet capturing and network analysis are CPU intensive tasks because of the high amount of data to process and copy. The utilities of jpcap and Winpcap libraries help increase performance of our analyzer. Below figures are standard figures depicting the advantage of WinPcap over other packet capturing approaches e.g. BSD- proposed approach for packet capture .Thus, as our monitor is using WinPcap, the efficiency would naturally increase.

Performance tests confirm the excellent performance of WinPcap. The packet generation process, even in presence of a strange behaviour (maximum number of packets per second is reached with 88-bytes packets) is highly optimized and it is quite easy to reach the maximum load allowed by a Fast Ethernet LAN. The capturing process also has an excellent implementation and it outperforms the original BPF / `libpcap` implementations.

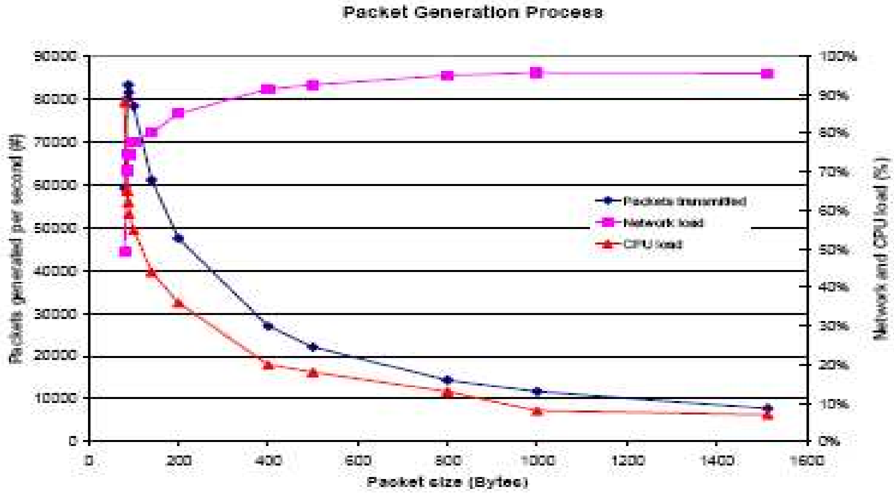
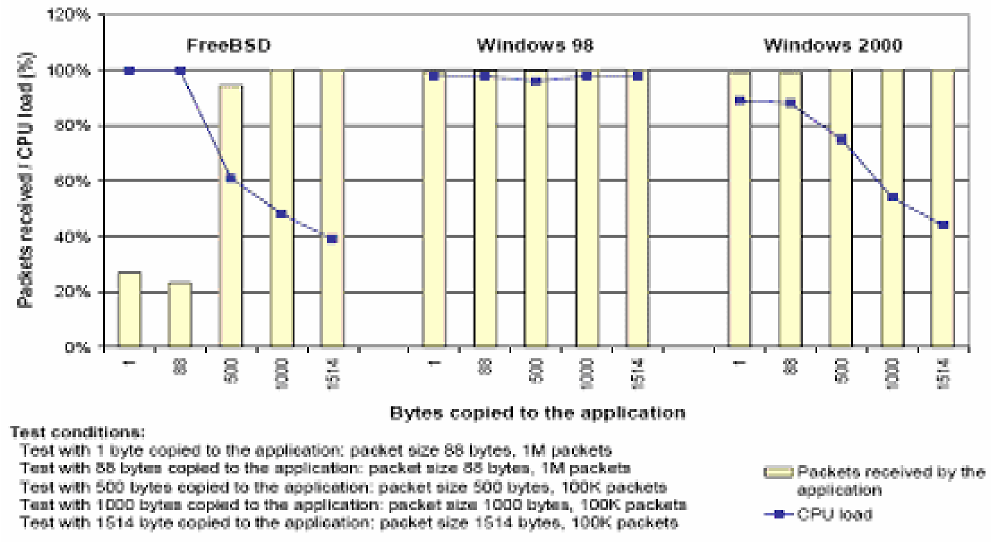


Figure 6. Depicting maximum number of packets per second and corresponding network load of WinPcap[26]



Test conditions:
Test with 1 byte copied to the application: packet size 88 bytes, 1M packets
Test with 88 bytes copied to the application: packet size 88 bytes, 1M packets
Test with 500 bytes copied to the application: packet size 500 bytes, 100K packets
Test with 1000 bytes copied to the application: packet size 1000 bytes, 100K packets
Test with 1514 byte copied to the application: packet size 1514 bytes, 100K packets

Figure 7. Delivering to the application performance[26].

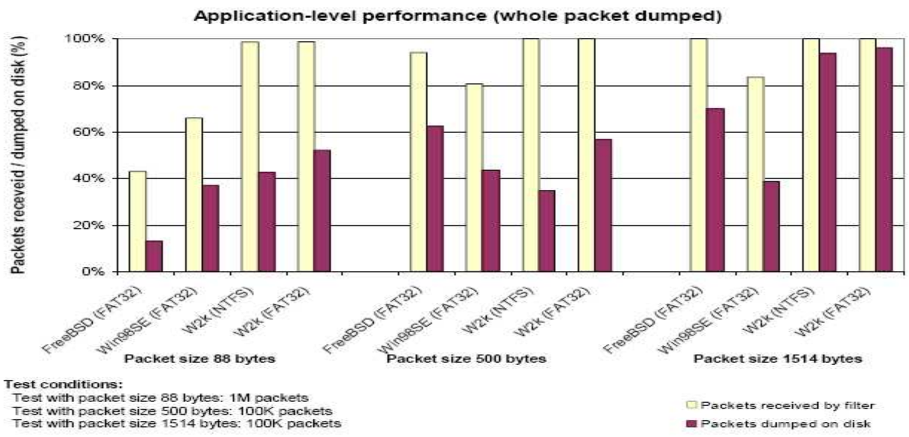


Figure 8. Performance of the complete capturing architecture, dumping the whole packet to file[26].

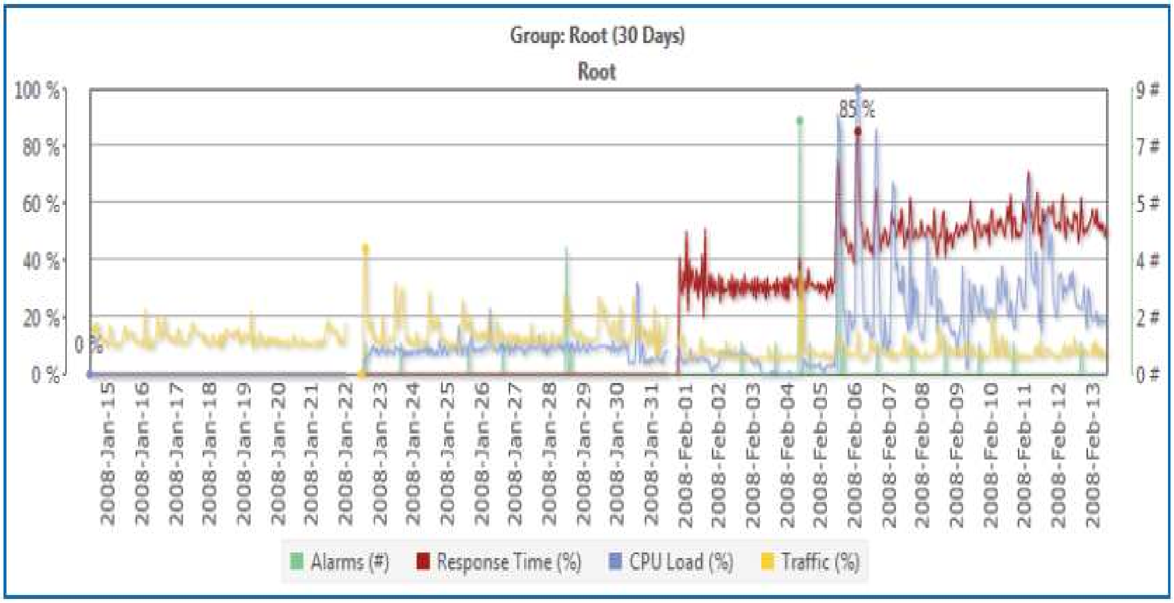


Figure 9. Showing an overview of PRTG Network Monitor’s aggregated statistics showing availability, bandwidth/traffic, CPU-load and alerts according to individually configured criteria

A few popular general purpose Sniffers are NAI Sniffer (commercial), Wireshark (previously know as Ethereal, an Open Source GUI Sniffer for Linux, Windows and other platforms), TCPDump (Open Source command line Sniffer for *nix - any Unix like operating system like Linux or FreeBSD-) and its Windows version called WinDump. Taking one such packet monitor like PRTG Network Monitor, its figure is shown below. This monitor covers all aspects of network monitoring from up-/downtime monitoring, traffic and usage monitoring, SNMP, NetFlow to packet sniffing.

7. Conclusion

Network monitor can be connected to the network to analyze what is happening inside the network. “Algorithms for Monitoring Data on Internet” is used to develop a log file of packet contents that are transferred to the user systems which is connected via a Local Area Network. The analysis of capturing packets that are traveling from the network to the user system has been analyzed by using the Jpcap, Winpcap open libraries, which provide java classes and functions for this purpose only. Using of Winpcap is essential to make the network monitor Windows compatible. WireShark and other commercially available network monitors have drawbacks like loss of large packets during transmission from the network. This drawback has been removed by using our proposed algorithm as it is based on using threads which functioning simultaneously capture packets (light-weight process). In our network analyzer we have used Java to create our network analyzer because of its features of object oriented programming (thus, better than C language), easy to use features and ability to create good GUI, graphical user interfaces (thus instead of C++).

In the proposed methodology, the packets traveling via the network to user node have been captured using Jpcap class functions and creating threads. Java supports multithreading or, JAVA is concurrent. Threads or, lightweight processes are lines if execution within program. Installing the software based on the proposed methodology, the system is aimed to act like an approximate network monitor in some contexts as it helps in evaluation on data flow from the network.

References

[1] Data Communications and Networking, *Fourth Edition* by Behrouz A Forouzan

[2] D. Comer (Ed.), Internetworking with TCP/IP: Principles, Protocols and Architecture, Prentice-Hall, Upper Saddle River, New Jersey, 1991.

[3] M. Allman, “A Web Server’s View of the Transport Layer,” *Computer Communication Review*, 30(5),

[4] Oct. 2000.

[5] http://jpcap.sourceforge.net/javadoc/index.html

[6] H. Balakrishnan, S. Seshan, M. Stemm, and R. Katz, “Analyzing Stability in Wide-Area Network

[7] Performance,” In *Proc. ACM SIGMETRICS’ 97*,June 1997.

[8] M. Crovella and A. Bestavros, “Self-similarity in World Wide Web Traffic: Evidence and Possible Causes,” *IEEE/ACM Transactions on Networking*, 5(6):835-846, December, 1997.

[9] "Internet RFC 2616.", section 1.4.

[10] Resource for Comments (RFC) Nos. – 1123 and 1122 of the IETF(Internet Engineering Task Force)

[11] M. Crovella and A. Bestavros, “Self-similarity in World Wide Web Traffic: Evidence and Possible Causes,” *IEEE/ACM Transactions on Networking*, 5(6):835-846, December, 1997.

[12] C. Cranor, T. Johnson, V. Shkapenyuk, and O. Spatschek. Gigascope: High performance network monitoring with a SQL interface. Sigmod 2002 demonstration, 2002.

[13] R. Buyya. PARMON: a portable and scalable monitoring system for clusters. Software - Practice and Experience, 30(7):723–739, 2000.

[14] D. Carney, U. Cetintemel, A. Rasin, S. B. Zdonik, M. Cherniack, and M. Stonebraker. Operator scheduling in a data stream manager. In VLDB, 2003.

[15] “tcptrace” [Online]. Available: www.tcptrace.org [Accessed: Sep. 2, 2009].

[16] “ethereal [Online]. Available: www.ethereal.com [Accessed: Sep. 7, 2009].

[17] “netwitness” [Online]. Available:www.netwitness.com [Accessed: Sep. 10, 2009].

[18] “snort”, [Online]. Available: www.snort.org [Accessed: Sep. 12, 2009].

[19] “paessler” [Online]. Available: www.paessler.com [Accessed: Sep.14, 2009].

[20] “networkworld” [Online]. Available: www.networkworld.com [Accessed: Sep. 18, 2009].

[21] “cisco” [Online]. Available: www.cisco.com [Accessed: Sep. 18, 2009].

[22] “techrepublic” [Online]. Available: www.articles.techrepublic.com.com/5100-10878_11-5815384.html [Accessed: Sep. 19, 2009].

[23] “ietf” [Online]. Available: www.ietf.org/internet-drafts/draft-ietf-httpbis-p1-messaging-05.txt [Accessed: Sep. 20, 2009].

[24] “jcap” [Online]. Available: www.netresearch.ics.uci.edu/kfujii/jpcap/doc/javadoc/index.html [Accessed: Sep. 20, 2009].

[25] “jcap” [Online]. Available: www.netresearch.ics.uci.edu/kfujii/jpcap/ChangeLog [Accessed: Sep. 20, 2009].

[26] “rongeek” [Online]. Available: www.irongeek.com [Accessed: Sep. 24, 2009].

[27] Fulvio Risso and Loris Degioanni , “An Architecture for High Performance Network Analysis”, Dipartimento di Automatica e Informatica – Politecnico di Torino Corso Duca degli Abruzzi, 2009.

Feature Parameter Extraction from Wavelet Sub-band Analysis for the Recognition of Isolated Malayalam Spoken Words

Vimal Krishnan V.R, Babu Anto P
School of Information Science and Technology
Kannur University, Kerala, India. 670 567
vimallnair@yahoo.co.in, bantop@gmail.com

Abstract*: The aim is to improve the recognition rate by finding out good feature parameters based on discrete wavelet transform techniques. The data set is created by using Malayalam spoken words which is collected from twenty individuals in various time intervals. We have employed Daubechies wavelet for the experiment. The feature vector was formed by using the parameters extracted from discrete wavelet transform techniques. The feature vector was produced for all words and formed a training set for classification and recognition purpose. Feature vectors of element size sixteen was collected for all the words by using classical wavelet t decomposition technique.*

Keywords: Wavelet, Speech Recognition, Feature Extraction, Artificial Neural Network.

1. Introduction

Over the past many decades the researchers are trying to come out with new feature parameters which give good recognition result for computer speech recognition. Majority of the research activities are focusing on some of the conventional transform techniques like FFT, MFCC, LPC and STFT etc. Speech signals from human are considered to be non stationary in nature. It is very difficult to analyze these non stationary signals by using these conventional transform techniques. The conventional transform techniques only focus on the frequency parameters extracted from the speech signal. Large variation in speech signals and other criteria like native accent and varying pronunciations makes the task very difficult. Scientists all over the globe have been working under the domain, speech recognition for last many decades. This is one of the intensive areas of research [1]. However automatic speech recognition is yet to achieve a completely reliable performance. Hence ASR has been a subject of intensive research. Recent advances in soft computing techniques give more importance to automatic speech recognition. ASR is a complex task and it requires more intelligence to achieve a good recognition result. In abstract mathematics, it has been known for quite some time that techniques based on Fourier series and Fourier transforms are not quite adequate for many problems. Wavelet based transform techniques remains indifferent in handling such problems. We have used wavelet based feature extraction for developing a feature vector. Performance of the overall system depends on pre-processing, feature extraction and classification. Selecting a feature extraction method and classifier often depends on the available resources. Wavelets are functions with compact support capable of representing signals with good time and frequency resolution. The choice of Wavelet Transform over conventional methods is due their ability to capture localized features [2]. ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. Here, accuracy has been increased by the combination of wavelet and artificial neural network.

The rest of the paper structured as follows: Section 2 gives a brief Review on Wavelet based feature extraction for speech recognition. Section 3 deals with the Classifier used for the Experiment. Section 4 discusses the creation of Speech database. Experiment and Results are summarized in section 5.

2. Feature Extraction Based On Wavelet

Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. The basic idea of the wavelet transform is to represent any arbitrary signal S as a superposition of a set of such wavelets or basis functions. These basis functions are obtained from a single photo type wavelet called the mother wavelet by dilation (scaling) and translation (shifts). The discrete wavelet transform for one-dimensional signal can be defined as follows [3].

$$c\,(a,b)=\int_R s(t)\,\frac{1}{\sqrt{a}}\,\Psi\,\frac{(t-b)}{a}\,dt \qquad (1)$$

The indexes c (a, b) are called wavelet coefficients of signal s(t), a is dilation and b is translation, Ψ(t) is the transforming function, the mother wavelet. It is so called because the wavelet derived from it analyzes signal at different resolutions (1/a). Low frequencies are examined with low temporal resolution while high frequencies with more temporal resolution. A wavelet transform combines both low pass and high pass filtering in Spectral decomposition of signals [3].

Wavelet analysis is a powerful and popular tool for the analysis of non stationary signals. The wavelet transform is a joint function of a time series of interest $x(t)$ and an analyzing function or wavelet $\tilde{A}(t)$. This transform isolates signal variability both in time *t*, and also in "scale" *s*, by rescaling and shifting the analyzing wavelet [4].

We have used wavelet based transform technique to extract feature from very complex speech data. Feature extraction involves information retrieval from the audio signal [5]. Here we have used Daubechies 4 (db4) type of mother wavelet for feature extraction purpose. Daubechies wavelets are the most popular wavelets. They represent the foundations of wavelet signal processing and are used in numerous applications. These are also called Maxflat wavelets as their frequency responses have maximum flatness at frequencies 0 and π.

2.1 Discrete Wavelet Transform

The transform of a signal is just another form of representing the signal. It does not change the information content present in the signal. For many signals, the low-frequency part contain the most important part. It gives an identity to a signal. Consider the human voice. If we remove the high-frequency components, the voice sounds different, but we can still tell what's being said. In wavelet analysis, we often speak of approximations and details. The approximations are the high- scale, low-frequency components of the signal. The details are the low-scale, high frequency components [6]. The DWT is defined by the following equation:

$$W(j,k) = \sum_j \sum_k x(k) 2^{-j/2} \psi(2^{-j} n - k) \tag{2}$$

Where (Ψt) is a time function with finite energy and fast decay called the mother wavelet. The DWT analysis can be performed using a fast, pyramidal algorithm related to multirate filter banks. As a multirate filter bank the DWT can be viewed as a constant Q filter bank with octave spacing between the centers of the filters. Each sub band contains half the samples of the neighboring higher frequency sub band. In the pyramidal algorithm the signal is analyzed at different frequency bands with different resolution by decomposing the signal into a coarse approximation and detail information. The coarse approximation is then further decomposed using the same wavelet decomposition step. This is achieved by successive highpass and lowpass filtering of the time domain signal and is defined by the following equations:

$$y_{\mathrm{low}}[n] = \sum_{k=-\infty}^{\infty} x[k] g[2n-k] \tag{3}$$

$$y_{\mathrm{high}}[n] = \sum_{k=-\infty}^{\infty} x[k] h[2n-k] \tag{4}$$

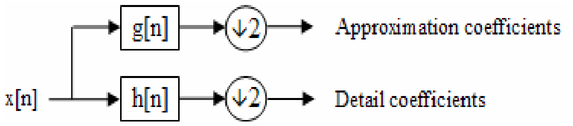


Figure 1: Signal x[n] is passed through low pass and high pass filters and it is down sampled by 2

$$y_{\mathrm{low}} = (x * g) \downarrow 2$$
$$y_{\mathrm{high}} = (x * h) \downarrow 2 \tag{5}$$

The Daubechies wavelets have surprising features, such as intimate connections with the theory of fractals. The peculiarity of this wavelet system is that there is no explicit function, so we can not draw it directly. What we are given is h(k)s, the coefficients in refinement relation which connect Ø(t) and translates of Ø(2t) these coefficients for normalized Daubechies -4 are as follows:

That is

Ø(t) = h(0)√2Ø (2t) + h(1)√2 Ø(2t-1) + h(2)√2 Ø(2t-2) + h(3) √2 Ø(2t-3)

(3)

Where Ø(t) is expressed in terms of Ø(2t) and its translates [7].

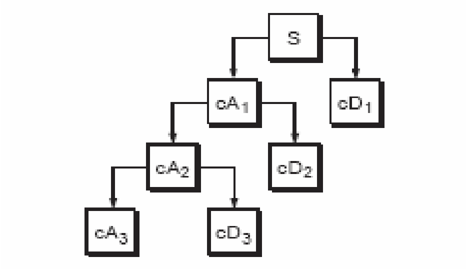


Figure 2: Decomposition Tree

3. Classification

In a general sense, a neural network is a system that emulates the optimal processor for a particular task, something which cannot be done using a conventional digital computer, except with a lot of user input. Optimal processors are sometimes highly complex, nonlinear and parallel information processing systems. Multi Layer Perception Network architecture is used for training and testing purpose. The MLP is a feed-forward network consisting of units arranged in layers with only forward connections to units in subsequent layers [8]. The connections have weights associated with them. Each signal traveling along a link is multiplied by its weight. The input layer, being the first layer, has input units that distribute the inputs to units in subsequent layers. In the following (hidden) layer, each unit sums its inputs and adds a threshold to it and nonlinearly transforms the sum (called the net function) to produce the unit output (called the activation). The output layer units often have linear activations, so that output activations equal net function values [8][9].

4. Selection of Data Set

For this experiment we have selected Malayalam language. We have used twenty Malayalam spoken words for the experiment. The International Phonetic Alphabet (IPA) format is shown in Table 1. We have selected words of a particular context. The selected words can be included in the category 'Consonant Vowel Consonant Vowel'. In Malayalam, no consonant is independent. It always stands with a vowel. The selected spoken words are very commonly used in Malayalam. Twenty words have been selected for the experiment. Samples were collected from twenty individuals in various time intervals. 32 speech samples were collected for each word. Six hundred and forty samples of twenty different words from twenty individuals are collected for the experiment.

Table 1: Input Data And Phonetic Alphabet

Sl. No	Words	In English	IPA
1	വേഗം	Vegam	//v//ɛ //ɡ //ɑ : //m/
2	പോയി	Poayi	/p//ɒ //eɪ //ɪ /
3	ചിരി	Chiri	/tʃ //ɪ //r//ɪ /
4	പാവ	Pava	/p//ɑ : //v//ɑ : /
5	വില	Vila	/ v//ɪ //l//ɑ : /
6	വീതം	Veetham	/v//ɛ //ɛ //θ//ɑ : //m/
7	നിറം	Niram	/n//ɪ //r//ɑ : //m/
8	വരം	Varam	/v//ɑ : //r//ɑ : //m/
9	പണം	Panam	/p//ɑ : //n//ɑ : //m/
10	നില	Nila	/n//ɪ //l//ɑ : /
11	പക	Paka	/p//ɑ : //k//ɑ : /
12	പാടം	Patam	/p//ɑ : //t//ɑ : //m/
13	നയം	Nayam	/ n//ɑ : //j//ɑ : //m/
14	പാലം	Palam	/p//ɑ : //l//ɑ : //m/
15	നാളെ	Nale	/n//ɑ : //l//ɛ /
16	മാരി	Mari	/m//ɑ : //r/ɪ
17	മൗനം	Maunam	/m//ɑ : //ʌ //n// ɑ : //m/
18	നിധി	Nidhi	/ n//ɪ // d// h//ɪ /
19	സമം	Samam	/s//ɑ : //m//ɑ : // m/
20	തരം	Tharam	/ θ//ɑ : //r//ɑ : //m/

5. Experiment and Result

Db4 type of wavelet is used in discrete wavelet decomposition technique. After conducting eighth level of decomposition we have collected largest and smallest elements from each sub band level. That is from each level of decomposition we have collected the maximum and the minimum values. The largest and smallest elements of each decomposition level are found to be the dominant feature value for each sample. These dominant elements are used to develop feature vector for each sample. Thus a feature vector of size sixteen is gained. This feature vector is given as an input to ANN classifier for the training purpose. A feature vector for testing purpose is also developed by using the same techniques. Fifteen samples for twenty Malayalam words are collected from different individuals and stored under various words categories. We trained the training set using the multi layer perceptron network architecture.

From the experiment we obtained a 84% recognition (Out of 640 speech samples 538 could be classified correctly) by using Artificial Neural Network. The results are shown in the figure 3 and 4.

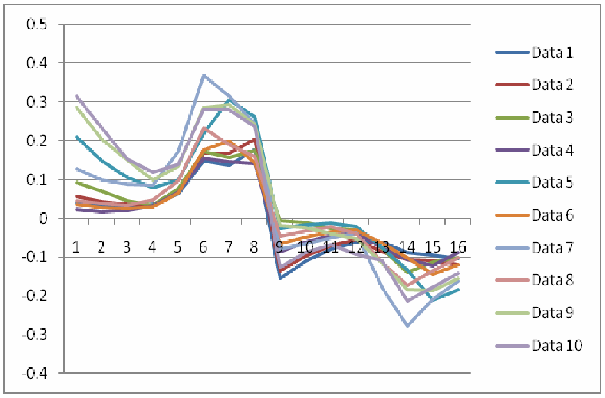


Figure 3. Graph plotted with the feature values of a speech data that is used for the experiment.

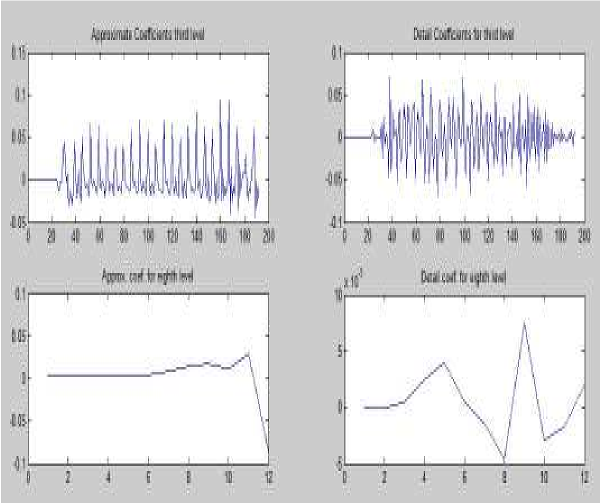


Figure 4. Various decomposition levels of sample speech data

6. Conclusion

From this study we could understand and experience the effectiveness of Classical Wavelet Decomposition. The performance of discrete wavelet decomposition in feature extraction is appreciable. We have also observed that, Neural Network is an effective tool which can be embedded successfully with wavelet. The efficiency of the method is to be verified with very large database.

Acknowledgments

The authors would like to thank Kerala State Council for Science, Technology and Environment, Govt. of Kerala, India for their support and grant awarded for the conduct of this research work.

Reference

[1] Lawrence Rabiner, Biing-Hwang Juang, Fundamentals of Speech Recognition, Englewood Cliffs, NJ: Prentice-Hall,1993.

[2] Mallat, Stephane, A wavelet tour of signal processing, San Diego: Academic Press, 1999. ISBN 012466606.

[3] G.K. Kharate1, A.A. Ghatol2 and P.P. Rege3 “Selection of Mother Wavelet for Image Compression on Basis of Image”, IEEE - ICSCN 2007, MIT Campus, Anna University, Chennai, India. Feb. 22-24, 2007 pp.281-285.

[4] Mallat, SA, Theory for multiresolution signal decomposition: The Wavelet Representation, IEEE Transactions on Pattern Analysis Machine Intelligence, vol.31, 1989, pp 674-693.

[5] K..P.Soman, K.I. Ramachandran, Insight into Wavelets from theory to Practice, Second Edition, PHI, 2005.

[6] Kadambe, S; Srinivasan, P, Application of Adaptive Wavelets for Speech, Optical Engineering 33(7, pp 2204-2211), July 1994.

[7] Stuart Russel, Peter Norvig, Artificial Intelligence A Modern Approach, New Delhi:Prentice Hall of India, 2005.

[8] S N Sivanadam, S Sumathi, S N Deepa Introduction to Neural Networks using Matlab 6.0 ,New Delhi: Tata McGraw-Hill, 2006.

[9] James A Freeman, David M Skapura, Neural Networks Algorithm , Application and Programming Techniques , Pearson Education, 2006.

Vimal Krishnan V R is working as Project Fellow under the domain Speech Processing at School of Information Science and Technology, Kannur University, India. In 2005, he has received his Master of Science degree in Software Science from the Periyar University, Tamil Nadu, India. He is Persuading his Doctoral degree in Kannur University. His main research interest lies in the area of Soft Computing Techniques, Speech and Signal Processing and Pattern recognition.

Babu Anto P is working as Head, Department of Information technology, Kannur University, India**.** He has received his Master of Science degree from Cochin University of Science and Technology, India in 1982 and he has awarded with his Doctoral Degree in 1992 by Cochin University. He has a number of international journals and conference papers in his credit. He is guiding doctoral students for the past years. His mail research interest lies in Speech processing, Pattern Recognition, Data mining and visual Cryptography.

Implementing Fuzzy-Genetic Algorithm in Mental Health Diagnostic Expert System

Rozita Yati Masri, Dr. Hajar Mat Jani, Alicia Tang Yee Chong
Department of Systems and Networking, College of IT,
Universiti Tenaga Nasional
KM 7, Jalan Kajang-Puchong,
43009 Kajang, Selangor
ozitay@uniten.edu.my, hajar@uniten.edu.my, aliciat@uniten.edu.my

Abstract: ***Fuzzy-Genetic Algorithm (fuzzy-GA) is used to calculate the risk probability of possible suitable sets of treatments within the proposed Mental Health Diagnostic Expert System. There are many different sets of treatments for a specific mental disorder. The price variation between the sets of treatments can sometimes even reach hundreds. It is obvious that the better the quality of the treatments, the more expensive the price gets. However, that perspective can be misleading, causing patients into wanting the more expensive set of treatments, when in reality, there may be other sets of treatments that give the patient the same effect but with lower price. Thus, Mental Health Diagnostic Expert System is proposed. It is designed to assist psychotherapists in diagnosing mental patients, and in coming up with options of suitable treatment plans for the patients based on the patients' condition and financial planning. Each set of mental treatments, which specifically involved medications, comes with risk (i.e. side effects). The probability of the risk to take effect depends on the patient's current condition. Hence, to calculate the risk probability of possible suitable sets of treatments, fuzzy-GA is applied, resulting in more optimized options of solutions.***

Keywords: mental treatments; Mental Health Diagnostic Expert System; fuzzy-Genetic Algorithm

1. Introduction

Due to the economic crisis, the number of people having to deal with stress increases. If these people, which of course the majority is either from the moderate or poor family, are not able to cope with their stress, they would be in danger of developing some sort of mental disorder(s). The levels of domestic violent and child abuse are increasing as the financial crisis pressure on, stressing out the ones who are responsible for the family [1]. As stressed out parents put too much pressure on their children, the children would also be in danger of developing some disorders related to depression or anxiety [1].

Already, more than 450 million people around the world are suffering from some kind of mental disorders. Mental disorders are actually more common compared to cancer, diabetes, or heart disease; yet the recovery rate is higher than many physical illnesses [2]. But even with a higher rate of getting recovered, many choose not to seek for professional treatment. Some of the reasons include [3]:

1. They fear that others will find out about the sickness.
2. They consider their sickness as embarrassing.
3. They cannot afford to pay for the treatments.
4. They hold a belief in the existence of witch-crafts or demonic possessions and the only way to cure the conditions is through exorcisms.
5. They do not know where to go and ask for help.
6. They are not aware that they are mentally ill.
7. They do not know that they should see a psychotherapist.
8. They prefer to seek help only from the people who are close to them.

The sets of treatments for mental disorders are very expensive. For the moderate and the poor who suffers from or cares for a family member with mental disorder(s), this may seriously have an effect on their financial planning. In an attempt to help these families in getting some kind of mental treatments that is within their financial budget, Mental Health Diagnostic Expert System is proposed. The Expert System (ES) will be operated by a psychotherapist, which will diagnose a patient's mental condition based on the symptoms, physical test results, and medical test results. After the ES has determined the condition of the patient, fuzzy-Genetic Algorithm (fuzzy-GA) will then be applied in determining the suitable treatment plans for the patient, based on the patient's condition and financial budget. We decided to use fuzzy-GA in determining the treatment plans due to the fact that not everyone can afford the best set of treatments since the best is almost always the most expensive. The fuzzy-GA component in the ES will assist the system to recommend more affordable alternative treatment plans suitable to the patient's condition and budget. It is better for the patients to get the alternative treatments rather than simply ignoring their problem by pretending that the sickness would go away naturally.

The main objective is to use fuzzy-GA as a technique in coming up with the suitable treatment plans for mental patients. The suitability of a treatment is measured based on the patient's condition and budget for the treatment. The whole purpose is to optimally reduce the price of mental treatments (if applicable) so that mental patients from the moderate and the poor family are able to get the help they need.

2. Research Contribution

As mentioned earlier, there are many sufferers of mental disorders who refuse professional help. The main reasons mostly involved the stigma on the issue of mental disorders itself. Another popular reason revolves around the affordability of getting the treatment for the mental condition. So, Mental Health Diagnostic Expert System is proposed, which benefits both the psychotherapists and the patients of mental disorders.

The ES will assist new and inexperienced psychotherapists to diagnose and treat a patient with effective consultation, as if a real expert is supervising them. As the ES is being used by more psychotherapists, the diagnoses and treatments will eventually become more standardized; therefore, indirectly will increase the accuracy of the diagnoses.

The patients then have the advantage in getting more accurate diagnoses and a lower risk of being misdiagnosed. The ES also helps in lowering the price of diagnoses sessions and the treatments, to fit within the patient's budget, as fuzzy-GA is used to derive several optimized solutions.

3. Literature Review

Genetic algorithms are a class of search probability algorithms based on biological evolution [4], which is inspired by the Darwin's theory of evolution, and is used in optimization problems. Biologically, every single living organism consists of cells, where in each cell is a set of chromosomes. In the ES, the problem variable domains are represented as the chromosomes of a fixed length [4]. The chromosomes in this case are represented by a set of solutions, called population, and the solutions from the population are then used to create new population with better solutions [5]. The general algorithm is as followed:

i. Choose the initial population
ii. Evaluate the fitness of each individual in the population
iii. Repeat the following steps until the termination criterion is satisfied:
   a. Select the best ranked individuals to reproduce
   b. Breed new generation through crossover and mutation
   c. Evaluate fitness of each individual offspring
   d. Replace the worst ranked part of the population with the offspring

This algorithm will be used as the guidelines in determining the suitable treatment plan(s). It is possible to have many different sets of treatment plans where the risk factors and prices vary. Fuzzy-GA will be used to help determine the optimized sets of treatment plans based on the conditions of the patients to ensure the risk involved is at the minimal level, and within the patient's budget.

The previous works focused on determining the type of mental disorder(s) a patient is suffering from, along with the severity of the disorder(s). The determination of the disorder(s) and the severity is based on the symptoms the patient has, the results of both physical and medical tests, and the duration or frequency of the symptoms. Rule-based reasoning and fuzzy logic are used in implementing the mental disorders determination part of the system.

Xpert4Health: Web Based Medical Protocol & Collaboration System has been used as an example and comparison in developing the proposed Mental Health Diagnostic Expert System [3]. It is a web-based health diagnostic ES, which focuses in liver disease and in and in particular Hepatitis-C. The system computerizes the decision making process and offers education and reinforcement for staffs. It also tolerates performance monitoring anywhere within the domain [6].

We have decided that the proposed Mental Health Diagnostic Expert System will have some similar features to Xpert4Health. The proposed ES will assist the psychotherapists in performing the assessments, and come up with the diagnosis, commentary and suggested treatment plans [7].

4. Knowledge Base

The knowledge base acts like the brain of the system. It stores all the important knowledge needed to be manipulated by the system. The knowledge base of the proposed ES is used to store all the data regarding the mental disorders and the sets of treatments. These data include (and not limited to): the names and types of disorders, the symptoms of the disorders, the names and types of medicines, the requirements of each medicine, and the approximate prices of the medicines. Rule-based reasoning and fuzzy logic are used in developing the knowledge base as it is "beneficial in assisting inexperienced psychotherapists in performing their job not only because the approach resembles human reasoning, but also because of the heuristic programming, which enables the ES to process the data fairly fast and more systematic[7]."

In selecting the suitable sets of treatment plans, the system will interact with the knowledge base frequently, as it retrieves the data it needs to manipulate and come up with the suitable sets of treatment plans for the patient.

5. Fuzzy-Genetic Algorithm

Assume that a patient is diagnosed using the proposed Mental Health Diagnostic Expert System, and it has been determined that the patient is suffering from Bipolar I Disorder, Most Recent Episode Depressed, with a moderate severity. To proceed in getting the suitable sets of treatment plans for the patient, the patient will then be asked to provide financial information of the amount he or she finds affordable.

The price range is then stored and will be used later to determine which of the sets of treatment plans fall within the patient's budget. The system will extract the type of treatments available for such disorder and severity, which in this case, include mood stabilizer and antidepressant. When the patient's condition is considered stabilized, only then will the psychotherapy is added as part of the treatment. Since the patient's disorder is not severe, neither Electroconvulsive Therapy (ECT) nor hospitalization is required.

Patient's current overall condition is then determined by the presence or absence of certain medical or physical conditions. Upon the presence of a certain condition, a value 1 is assigned to such condition; whereas a value -1 is assigned to the conditions absent from the patient's overall condition. From here on, fuzzy-GA will take place.

First, the initial population is determined (a group of treatment plans) of size N. The number of chromosomes is based on the number of options of treatment plans available for the disorder, after eliminating medicines that are unsafe for the patient.

Next, the evaluation of the fitness of each chromosome takes place. The lower the price of a low risk medicines within the budget, the better the fitness of the chromosome. If there is no low risk medicines within the budget, then consider higher risk medicines within the budget. If none, then some of the lowest prices medicines outside of the budget are considered.

After that, the average fitness of the population by summing up the fitness of all chromosomes in the population is determined, and the total is then divided by the number of chromosomes. The higher the average, the better the overall fitness is.

Mutation rate of 0.001 is used to represent the complexity of a human's condition, and eliminates an infinite loop. The crossover point is in the middle of each chromosome, due to the fact that each chromosome is representing two separate medicines. The offspring is then stored in a temporary storage to create the new population.

The mating process looping continues until the size of the new population reaches the size of the initial population. The evaluation of the fitness of each chromosome, again, takes place—this time, the evaluation is performed on the offspring of the previous chromosomes. If the last average fitness is worse than the new average, then it will replace the current population with the new population stored in the temporary storage. Then the temporary storage is cleared, so it can be used in creating another new population. It will then repeat the mating process looping again until the last average fitness is better than or equal to the new average. When that target is reached, the best five of the chromosomes stored in the temporary storage (in term of their fitness) will be displayed, where they represent the top five options of sets of treatments.

Figure 1 presents the flowchart of the fuzzy-GA design, as described above.

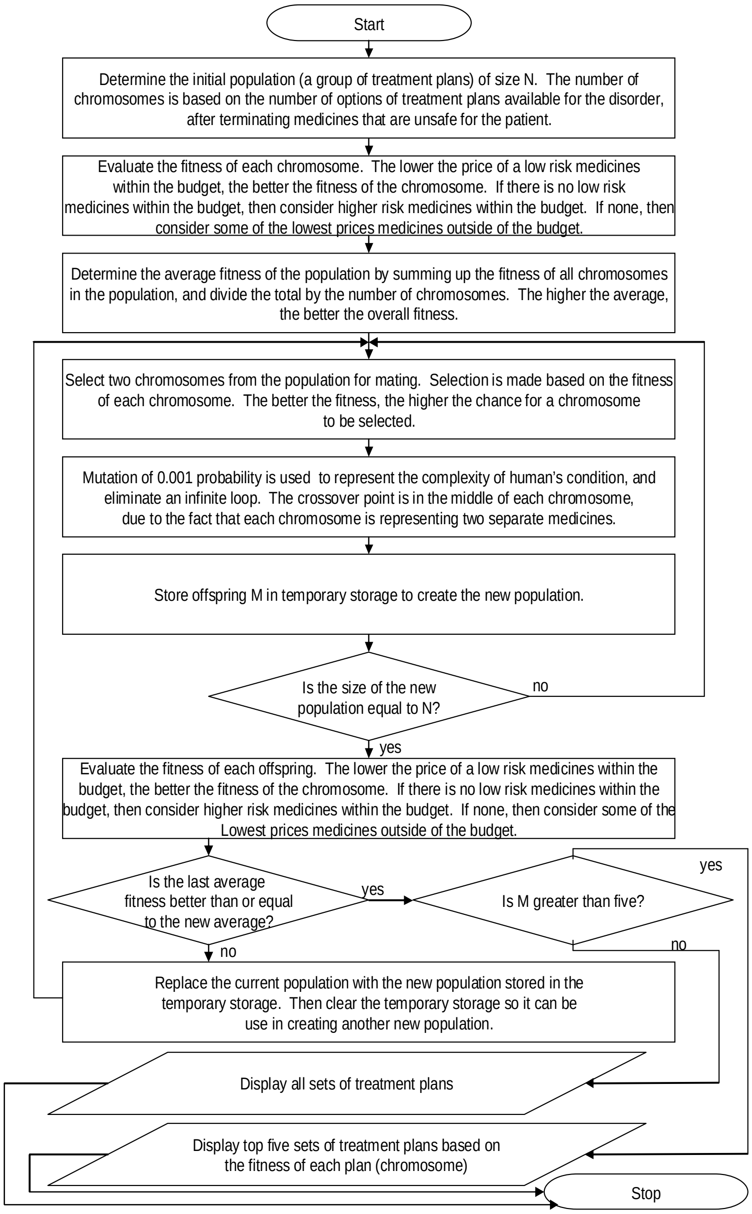

Start
Determine the initial population (a group of treatment plans) of size N. The number of chromosomes is based on the number of options of treatment plans available for the disorder, after terminating medicines that are unsafe for the patient.
Evaluate the fitness of each chromosome. The lower the price of a low risk medicines within the budget, the better the fitness of the chromosome. If there is no low risk medicines within the budget, then consider higher risk medicines within the budget. If none, then consider some of the lowest prices medicines outside of the budget.
Determine the average fitness of the population by summing up the fitness of all chromosomes in the population, and divide the total by the number of chromosomes. The higher the average, the better the overall fitness.
Select two chromosomes from the population for mating. Selection is made based on the fitness of each chromosome. The better the fitness, the higher the chance for a chromosome to be selected.
Mutation of 0.001 probability is used to represent the complexity of human's condition, and eliminate an infinite loop. The crossover point is in the middle of each chromosome, due to the fact that each chromosome is representing two separate medicines.
Store offspring M in temporary storage to create the new population.
Is the size of the new population equal to N?
no
yes
Evaluate the fitness of each offspring. The lower the price of a low risk medicines within the budget, the better the fitness of the chromosome. If there is no low risk medicines within the budget, then consider higher risk medicines within the budget. If none, then consider some of the Lowest prices medicines outside of the budget.
Is the last average fitness better than or equal to the new average?
yes
Is M greater than five?
yes
no
no
Replace the current population with the new population stored in the temporary storage. Then clear the temporary storage so it can be use in creating another new population.
Display all sets of treatment plans
Display top five sets of treatment plans based on the fitness of each plan (chromosome)
Stop

Figure 1. Flowchart of fuzzy-GA design

The population of chromosomes is determined by the number of options of medicines available for the patient's disorder. The set of treatments for a patient who suffers from Bipolar I Disorder, Most Recent Episode Depressed, consists of a combination of a mood stabilizer and an antidepressant. Assume that there are two types of mood stabilizers: Priadel (lithium) and Tegretol (Carbamazepine); and three types of antidepressants: Luvox (Fluoxetine), Prozac (Fluoxetine), and Zoloft (Sertraline). The ES will call each of the above from the knowledge base to extract the assigned values given based on the individual medicine's *contraindications* and *special precautions*.

The value of the patients' overall condition is formulated after the condition is gathered as mentioned earlier. Each individual medical and physical condition represents a digit to the value of the patient's overall condition.

The values assigned to the medicines are the same in length as the values given to the patient's overall condition because each digit assigned to the medicines corresponds sequentially to the conditions which made up the patient's overall condition.

To determine which of the medicines can be given to the patient, each medicine needs to be evaluated. The suitability is determined by the lack of conditions met for *contraindications* and *special precautions* of each medicine. Meeting a condition of any contraindication of a medicine means that the medicine is not suitable and should definitely not be given to the patient. As for the *special precautions*, the more conditions are met, the higher the risk for the patient if he or she is given such medicine.

To examine the suitability, simply compare each digit of the assigned value for each medicine to the corresponding digit of the value for patient's condition, and come up with the overall values (see Figure 2).

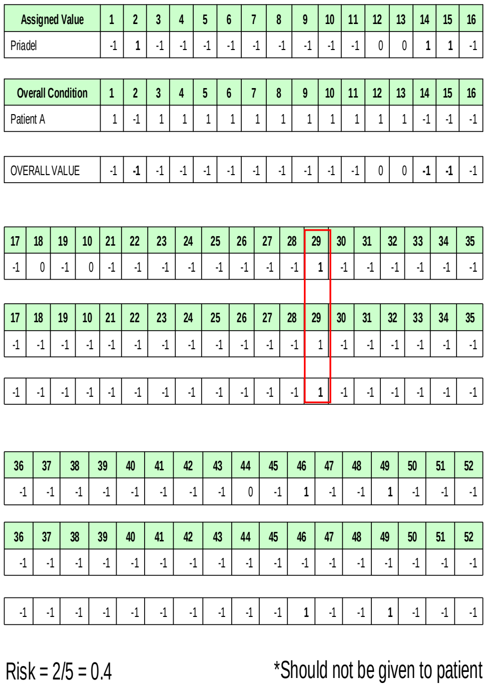
The rule of comparison leading to the overall value is as followed:

- Take the smaller value between the two digits being compared as the parallel digit of the overall value.

The logic behind is simple: if a digit has a value of -1 for the medicine (no threat to patient even if the condition exists), and the corresponding digit from the patient's condition is 1 (the condition does exist), the overall value is still a -1 because the medicine does not threaten the patient's condition. If a digit has a value of 0 (possible threat) and the patient's condition is a -1 (patient does not meet such condition), then the overall value is still a -1. If a digit has a value of 1 (definite threat) and the patient's condition is a -1, the medicine is still not a threat to the patient because the patient does not meet the threatening condition.

In order for a medication to be a threat to a particular patient's condition, both values of corresponding digits have to be true (which is represented by the value 1), thus the overall value is 1. If a digit has a value of 0 for the medicine, it means that there is a possible risk for the patient (depending on the patient's condition). So if the patient doesn't have such condition, then there will not be such risk for the patient, leaving the overall value at -1. If such condition does exist however, then there would be a risk for the patient, therefore the overall value will be 0.

The medicines are then evaluated based the existence of any value 1 and the number of value 0 in the overall values. If there is any value 1 exists in the overall values, then such medicines should not be given to the patient due to the threat it would cause (exemplified in Figure 2). The risk of each medication is calculated by the number of value 0 in the overall values divided by the number of value 0 in the assigned value of the medicine. The possibility of risk increases as more value 0 is produced in the overall values.



Assigned Value	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Priadel	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	1	1	-1

Overall Condition	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Patient A	1	-1	1	1	1	1	1	1	1	1	1	1	1	-1	-1	-1

OVERALL VALUE	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1

17	18	19	10	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
-1	0	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1

17	18	19	10	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1

36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
-1	-1	-1	-1	-1	-1	-1	-1	0	-1	1	-1	-1	1	-1	-1	-1

36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

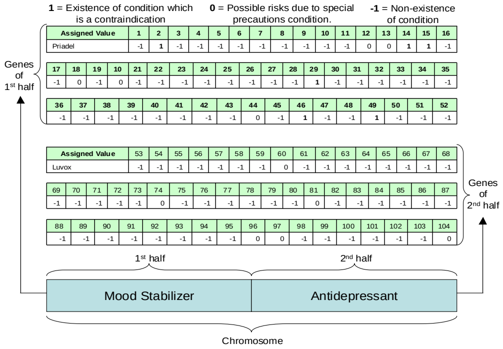
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	1	-1	-1	-1

Risk = 2/5 = 0.4 *Should not be given to patient

Figure 2. Evaluation of medicine in determining whether or not the medicine can be given to the patient

After eliminating the medicines that should not be given to the patient, it is time to determine the population for the genetic algorithm to take place. The population is made out of numerous sets of combined medications of mood stabilizers and antidepressants as the chromosomes.

Each chromosome in this example is made up of 104 digits or genes (See Figure 3), combining the assigned values for mood stabilizers and antidepressants. The first half of each chromosome (the first 52 genes) represents the *contraindications* and *special precautions* of the mood stabilizers, where each digit is of either -1 (which represents non-existence of threat to patient's condition), 0 (possible existence of risk to patient's condition—determined by the *special precautions*), or a 1 (definite existence of threat to patient's condition—determined by the contraindication of the medicine). The second part of the chromosome (the last 52 genes) represents the *Contraindications* and *special precautions* of the antidepressants, similar to the first 52 genes.


1 = Existence of condition which is a contraindication
0 = Possible risks due to special precautions condition.
-1 = Non-existence of condition
Genes of 1st half
Genes of 2nd half
1st half
2nd half
Mood Stabilizer
Antidepressant
Chromosome

Assigned Value	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Priadel	-1	**1**	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	**1**	**1**	-1

17	**18**	**19**	**10**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**
-1	0	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	**1**	-1	-1	-1	-1	-1	-1

36	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**
-1	-1	-1	-1	-1	-1	-1	-1	0	-1	**1**	-1	-1	**1**	-1	-1	-1

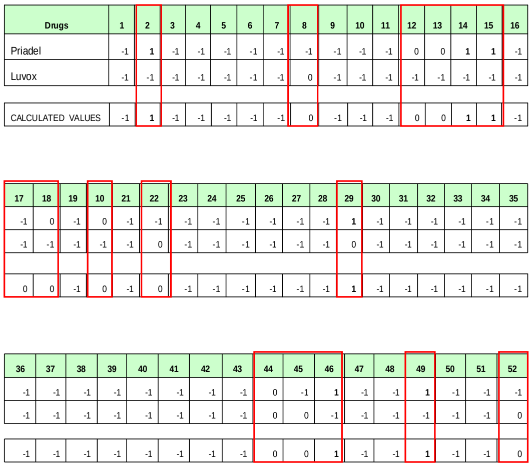
Assigned Value	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68
Luvox	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1

69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1

88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	0

Figure 3. Illustration of each chromosome

To determine the fitness of each chromosome, the first 52 genes of the chromosome are compared with the last 52 genes to come up with the calculated value, by taking the greater value of the two corresponding digits (see Figure 4). The calculated value is then compared to patient's overall condition by taking the lesser value of the corresponding digits (similar as to when eliminating the unsuitable medicines), and the risk is then calculated by the number of value 0 in the overall values divided by the number of value 0 in the assigned value of the medicine. Table 1 illustrates the certainty factor of risk probability.



Drugs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Priadel	-1	**1**	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	**1**	**1**	-1
Luvox	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1
CALCULATED VALUES	-1	**1**	-1	-1	-1	-1	-1	0	-1	-1	-1	0	0	**1**	**1**	-1

17	**18**	**19**	**10**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**
-1	0	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	**1**	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	-1
0	0	-1	0	-1	0	-1	-1	-1	-1	-1	-1	**1**	-1	-1	-1	-1	-1	-1

36	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**
-1	-1	-1	-1	-1	-1	-1	-1	0	-1	**1**	-1	-1	**1**	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	0
-1	-1	-1	-1	-1	-1	-1	-1	0	0	**1**	-1	-1	**1**	-1	-1	0

Figure 4. Determining the calculated value

Table 1: Fuzzy Certainty Factor

Value of parameter	Fuzzy Term	Fit/Unfit
0.0	Definitely not	Fit
0.1	Almost certainly not	Fit
0.2	Probably not	Fit
0.3	Maybe not	Fit
0.4	Unknown	Depends on result of mutation
0.5	Unknown	Depends on result of mutation
0.6	Unknown	Depends on result of mutation
0.7	Maybe	Unfit
0.8	Probably	Unfit
0.9	Almost certainly	Unfit
1.0	Definitely	Unfit

The range of risk probability is 0.0 to 1.0, and is assigned to a certainty factor of fuzziness (see Table 1). Risk probability of 0.0 until 0.3 is considered low and therefore is measured as fit. Risk probability of 0.7 and higher is measured as unfit. A 0.4, 0.5, or 0.6 is considered as within the unknown zone. If the probability is 0.4, then it will be set to 0.3, while if the probability is 0.6, it will be set to 0.7; measuring it as either fit or unfit to be given to the patient. At the risk probability of 0.5, the setting is done at random, where the probability can be set to either 0.3 or 0.7.

Note that unfit does not mean it cannot be given to the patient because patient's conditions do not clash with the *contraindications* of the medicine. It simply means that the risk is high, and therefore is not favorable. But if the favorable ones are much more expensive, and are not within the patient's budget, then the not so favorable ones will be considered.

Two chromosomes are then selected for crossover based on the fitness of each chromosome. The better the fitness of a chromosome, the higher the chance of it to be selected for crossover. As shown in Figure 5, the crossover point at the middle of each chromosome because each chromosome is made up of two different medicines.

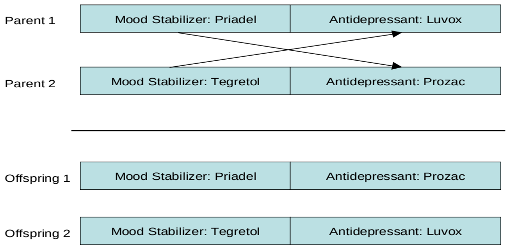


Figure 5. Crossover between two chromosomes

Mutation is set to take place after the crossover with a probability of 0.001. The mutation is needed since it represents the complexity and complication of human's condition. Aside from that, mutation also eliminates the possibility of infinite loops.

After the crossover, the offspring will be evaluated for their fitness, the same way as their parents had been evaluated. If the prices of the offspring still fall within the range of patient's budget, then both parents and the offspring will be stored, and another selection of parents will be crossed over as the next generation continues. The terminating condition of the loop is either when the prices of the parents are not within the range of the patient's budget, or when the average fitness of the new population is higher than the average of the previous population.

Price comparison is done based on the budget set by the patient and the minimum approximation of the price of medications.

6. Discussion

The suitable sets of treatment plans are derived based on the requirements of the medications as well as the requirements of the patient's condition. The value of each gene inside a chromosome is crucial as it helps in determining the fitness of each medicine based on the *contraindications* and the *special precautions*.

Simply selecting the safest type of medicines or the cheapest set of treatment plans does not equal to the best set of treatment plans for a patient. The safest type of medicines is usually very expensive, and the cheapest of the medicines usually imposes a higher risk for the patient. With the existence of crossover in the algorithm, each combination of medications is explored thoroughly, thus producing more options suitable for the patient.

7. Conclusion and Future Work

With the proposed Mental Health Diagnostic Expert System, many benefits will be gained by both mental health-related organizations and the society. This expert system will facilitate the new and inexperienced psychotherapists to diagnose and treat their patient more accurately. The diagnoses and treatments will eventually become standardized when more psychotherapists use the proposed ES. The treatments will be more affordable to more patients since the ES will derive the treatment plans based on the conditions of the patients, the prices of the treatments, and from the patients' financial budget. Due to the affordability of the treatments, more and more mental disorders sufferers will seek professional help, thus, lowering the percentage of untreated mental cases.

In reality, the system should also consider the drugs interactions. As part of the ongoing work and future word, we will include drugs interactions of each medicine in the Mental Health Diagnostic Expert System.

References

[1] "2008 Financial Crisis Affects Parenting Stress". Article Trader. Extracted on May 14th, 2009. Available at: http://www.articletrader.com/ health/stress/2008-financial-crisis-affects-parenting-stress.html

[2] "Startling Statistics About Mental Illness". A Source of Hope for All (ISHA International). Extracted on January 28th, 2009. Available at: http://www.myasha.org/node/12

[3] Rozita Yati Masri, Hajar Mat Jani, and Alicia Tang Yee Chong, "Expert System Approach in Diagnosing Mental Health: A Proposal". In Proceedings of the International Symposium on Information Technology (ITSim). 26-29 August 2008, Kuala Lumpur Convention Centre, Malaysia, pp. 261-266.

[4] Negnevitsky, Michael. "Artificial Intelligence: A guide to Intelligent Systems, Second Edition". Pearson Education Limited. 2005, pp.222.

[5] Orbitko, Marek. "Genetic Algorithms". 1998. Date of extraction: 18 January 2008. Available at: http://cs.felk.cvut.cz/~xobitko/ga/

[6] Terida Systems. "Xpert4Health: Web Based Medical Protocol & Collaboration System". Extrated on 31 March 2008. Available at: http://www.knowledgestorm.com/ActivityServlet?ksAction=optInReq&solId=58273&pos=1&referer=SEARCH_RESULTS&trkpg=search_results_solname

[7] Rozita Yati Masri, Hajar Mat Jani, and Alicia Tang Yee Chong, "Applying Artificial Intelligence Techniques to Mental Health Diagnostic Expert System", In Proceedings of the 4th International Conference on Information & Communication Technology and Systems, August 5 2008, Surabaya, Indonesia, pp. 375 – 382.

An Economical Model for Optimal Distribution of Loads for Grid Applications

G. Murugesan[1] and Dr.C. Chellappan[2]

[1]Research Scholar, Department of Computer Science and Engineering
Anna University, Chennai 600 025, India
murugesh02@gmail.com

[2]Professor and Head, Department of Computer Science and Engineering
Anna University, Chennai 600 025, India
rcc@annauniv.edu

Abstract: ***Resource allocation is an important component of a Grid Computing infrastructure. In this paper, we proposed a mathematical model for resource allocation for multiple Grid Applications with multiple processors (sinks). Most of the researches are based on the Divisible Load Theory mechanism; the processors which are participated in the processing have to select a divisible job and also all the processor has to complete the process at the same time. But in our work, initially considering the entire load from sources are divided into equally and select a set of processor from the available processors. The numbers of processors are selected with the help of its processing capacity. Equal allocation of load is attractive in multiple processor systems when real time information on processor and link capacity that is necessary for optimal scheduling is not available. A new mathematical model for minimizes the computing cost with equal allocation of divisible computation and communication load is developed. A cost optimal processor sequencing result is found which involves assigning load to processors in order of the cost per load characteristic of each processor.***

Keywords: Grid Scheduling, Integer Programming, Multiple Sources, Resource Allocation.

1. Introduction

Until very recently, grid computing was popular only in the research arena. However recent business reports suggest that enterprises are working towards implementing enterprise wide grids to share and utilize their vastly distributed computing resources. The aim of Grid computing is the utilization of underutilized resources to achieve faster job execution time. Gird can also provide access to software, computers, data, and other computing resources. Grid computing reduces the cost by connecting different machines like PCs and workstations to behave as a larger computational machine rather than purchasing special expensive machines to execute the complicated jobs. This can be sometimes showed as dividing the cost of the resources participating in the Grid between the users who use the Grid.

The management is an essential part in any system in the world. It comprises both hardware and software management. It determines if the system succeeds or fails. Furthermore, the Grid's resources are scattered and geographically distributed, therefore more care is needed when they are being managed. An important problem that arises in the area of grid computing is one of the optimally assigning jobs to resources to achieve business objectives. In Grid Computing environment Scheduling, performance prediction and resource management are important but challenging tasks. Grid scheduling consists of finding a suitable assignment between a computational workload and computational resources which are participated in the Computation. Reasonably allocating the resources to the tasks can effectively improve the concurrency ability of the distributed system. Over the past 19 years a good deal of research has been conducted on scheduling and load distribution with divisible loads. A divisible load is a data parallel load that can be arbitrarily partitioned among links and processors to gain the advantage of parallel processing. However most of this research has involved load distribution from a single source. That is, load originates from a single node in a larger grid or network. Multi-source load scheduling has received less attention but is a logical next step for research in this area.

Task scheduling is an integrated part of parallel and distributed computing. The Grid scheduling is responsible for resource discovery, resources selection, job assignment and aggregation of group of resources over a decentralized heterogeneous system; the resources belong to multiple administrative domains. The resources are requested by a Grid application, which use to computing, data and network resources etc. However, Scheduling an applications of a Grid system is absolutely more complex than scheduling an applications of a single computer. Because to get the resources information of single computer and scheduling is easy, such as CPU frequency, number of CPU's in a machine, memory size, memory configuration and network bandwidth and other resources connected in the system. But Grid environment is dynamic resources sharing and distributing. Then an application is hard to get resources information, such as CPU load, available memory, available network capacity etc. And Grid environment also hard to classify jobs characteristic, that run in Grid. There are basically two approaches to solve this problems, the first is based on jobs characteristic and second is based on a distributed resources discovery and allocation system. It should optimize the allocation of a job allowing the execution on the optimization of resources. The scheduling in Grid environment has to satisfy a number of constraints on different problems.

This paper is organized as follows: section 2 presents the related works, grid resource allocation is discussed in the section 3, section 4 shows the resource model, design and analysis of load distribution model described in the section 5 section 6 presents the mathematical model for resource allocation, section 7 shows the experimental results, section 8 and section 9 shows the conclusion and references respectively.

2. Related Works

In this section, we present some of the works that are relevant to the problem addressed in this paper. The problem of minimizing the processing cost of extensive processing loads originating from various sources presents a challenging task that if successfully met could faster a range of new creative applications. Inspired by this challenge, we sought to apply linear programming technique to assign equally divided jobs to the available resources. For divisible loads, research since 1988 has established that optimal allocation /scheduling of a divisible load to processors and links can be solved through the use of very tractable linear model formulation, referred in divisible load theory (DLT). DLT can model a wide variety of approaches with respect to load distribution (sequential or concurrent), communication (store and forward and virtual cut-through switching), and hardware availability (the presence or absence of front-end processors). Front-end processors allow a processor to both communicate and compute simultaneously by assuming communication duties [1]. DLT has been proven to be remarkably flexible. The DLT model allows analytical tractability to derive a rich set of results regarding several important properties of the proposed strategies and to analyze their performance. A 2002 paper on multi-source load distribution combining Markovian queuing theory and divisible load scheduling theory is discussed in Ko and Robertazzi [3]. In 2003 Wong, Yu, Veeravalli, and Robertazzi examined multiple source grid scheduling with buffer and without buffer capacity constraints [2]. Moges, Yu and Robertazzi considered multiple source scheduling using two root processor for small size models via linear programming and closed form solutions in 2004 and 2005, respectively [5], [6]. Marchal, Yang, Casanova, and Robert in 2005 studied the steady-state multi-application scheduling problem use of linear programming to maximize throughput for large grids with multiple loads/sources expresses a notion of fairness between applications [4]. Yu and Robertazzi proposed the use of min cost flow and multi-commodity flow formulations for steady state divisible load scheduling with multiple sources [10]. Viswanathan et al. [9] proposed a distributed algorithm to handle large volumes of computationally intensive arbitrarily divisible loads submitted for processing at cluster/ grid systems involving multiple sources and sinks. The different scheduling algorithms on heterogeneous platforms for divisible workloads are proposed by Beaumont et al [8]. This paper is significant for proposing some new optimization approaches for the multiple source scheduling problems in grids. Scheduling is one of the most studied topics in distributed systems. In this paper we dealt with multiple sources with multiple resources part of the work is based on optimal allocation of loads [7]. Each source having their jobs and the entire job of the each source is divided and each portion of the job has to be submitted from a set of sources. Initially we are selecting a set of best resources from the available resources. Each sources job has to be equally divided in to sub jobs and each sub job was assigned to a resource from the selected group of the resource.

3. Grid Resource Allocation

A generic grid computing system infrastructure considered hare comprises a network of supercomputers and/or a cluster of computers connected by local area networks, as shown in Figure. 1 having different computational and communication capabilities. We consider the problem of scheduling large-volume loads (divisible loads) within a cluster system, which is part of a grid infrastructure. We envisage this cluster system as a cluster node comprising a set of computing nodes. Communication is assumed to be predominant between such cluster nodes and is assumed to be negligible within a cluster node. The underlying computing system within a cluster can be modeled as a fully connected bipartite graph comprising sources, which have computationally intensive loads to be processed (very many operations are performed on them) and computing elements called sinks, for processing loads (data). To organize and coordinate distributed resources participating in the grid computing environment, we utilize the resource model which contain more than one sources and resources, based on which the uniform and scalable and allocating computational resources are identified. The contribution of this paper is that we propose equal division of load from a resource with the help of generating a random number and the divisible load is assign to a set of resources from the resources participated in the scheduling process. Now, we shall formally define the problem that we address.

4. Resource Model

Our model organizes sources and the resources with a structure of a hybrid hierarchical tree Figure 1. In the tree first level nodes called root nodes represent the sources which contains the divisible loads, the second level specifies the scheduler, which get the total workload (divisible load) from each sources, the third level specifies the equal division of loads of each sources, and the fourth level nodes (leaf nodes) are represented as resources or processors to perform the processes. In a dynamic Grid environment, nodes may join or leave frequently, and nodes status may change dynamically. In our model, we are considering the environment as static; i.e. once the node joined in the scheduling process the entire node has to participate in the processing. But in divisible load theory all the nodes has to involve or process all sources portion of jobs. In this model we are not compelling all the resources to process all the sources loads, instead only the selected set of resources

5. Design and Analysis of Load Distribution Model

In all the literature within the divisible load scheduling domain so far, an optimality criterion that is used to derive an optimal solution is as follows. It states that in order to obtain an optimal processing time, or an optimal make-span it is necessary and sufficient that all the sinks that participate in the computation must stop at the same time instant from a source. In multiple source environment all the resources has to take a portion of load from each sources as well as they has to be finish the work at the same time. In real time environment we cannot enforce all the processor to complete its processing at the same time. In our model we are not enforcing its load processing of all the processors at the same time as well as not all the processor has to take a portion of loads from all the sources.

Otherwise, load could be redistributed to improve the processing time. We use this optimality principle in the design of our distribution strategy. The scheduling strategy involves the portioning and distribution of the processing loads originated from S_1, S_2, … S_m. The total loads of the each source are equally divided in to portion of loads and each portion is allotted into a separate processor. Before job scheduling, the scheduler identifies the number of resources required to participate the process with respect to the numbers portions.

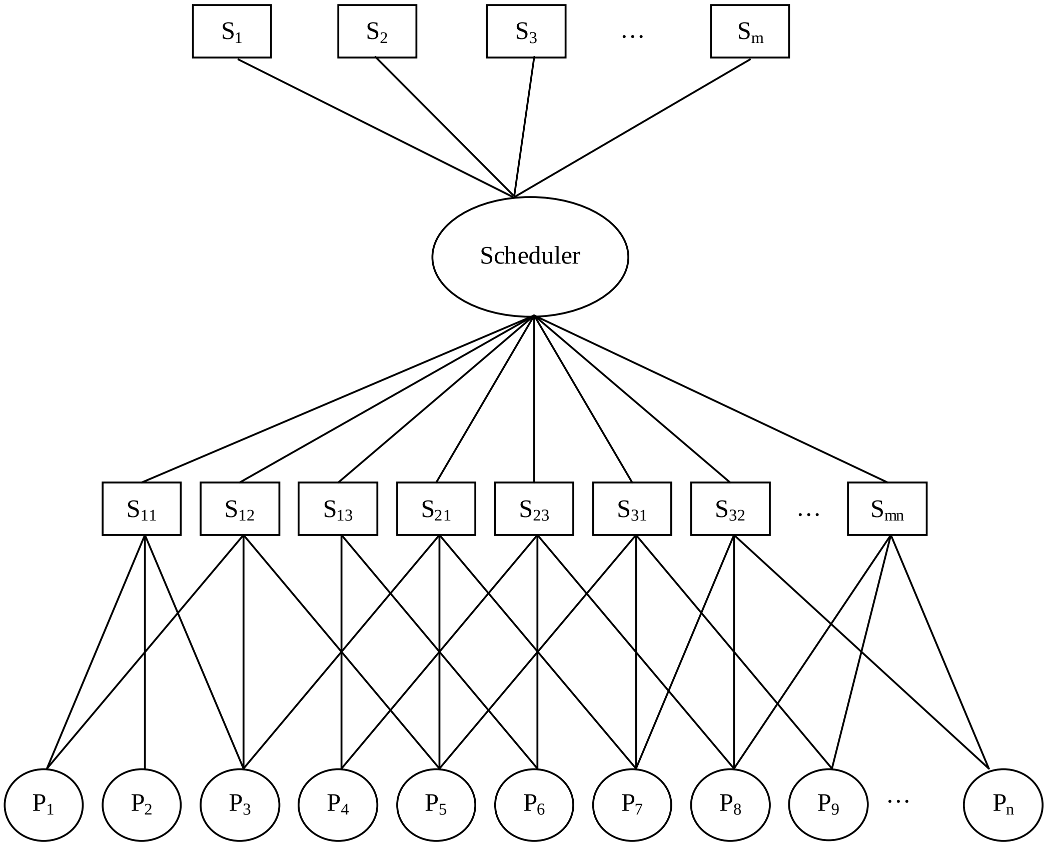


Figure1. Abstract view of a cluster comprising sources and sinks with a scheduler

We consider a tightly coupled, bipartite multiprocessor system. In the Grid system, we assume that there are M sources denoted as S_1, S_2, … S_m and N resources denoted as P_1, P_2, . . . P_n For each source, there is a direct link to all the resources. Each source Si has a load, and which is equally divided into portion of loads, denoted by S_{11}, S_{12}, . . . S_{nm}. Without loss of generality, we assume that all sources can send their loads to all the selected resource simultaneously. Similarly, we also assume that all the resource can receive load portions from all sources at the same time instant. The objective in this study is to schedule all the M loads among N resources such that the processing time, defined as the time instant when all loads have finished being processed by all the N resource, is minimal. The scheduling strategy is such that the scheduler will first obtain the information about the size of the loads that other sources have in their local memory. The scheduler will then calculate and notify each source of the optimum amount of load that each source has to give to each sink. This information can be easily communicated via any means of standard or customized communication protocol and it would not incur any significant communication overhead. The resources will then start computing the loads immediately after they receive their respective loads. It may be noted that we assume that each resource has adequate memory/buffer space to accommodate and process all the loads it receives from all the sources. We also assume that communication time is faster than computation time so no processor starves for load.

6. Notations Used In The Model

The Following symbols are used to define our mathematical model.

N No. of Processor required to complete the entire job of the grid user

M Total no. of Sources involved in the scheduling process

C_kAmount to spend to utilize the kth resource

T_k Time required to process a unit load by the kth resource

s_{ij} jth portion of load from ith source

b_i Budget of the ith source

T_kTotal available time of kth resource

x_{ijk} = 1 ; if ith source jth portion of load processed by kth resource

= 0 ; Otherwise

Minimize

$$\sum_{i=1}^{M}\sum_{j=1}^{m_i}\sum_{k=1}^{N} C_k t_k S_{ij} X_{ijk}$$

Subject to

$$\sum_{i=1}^{M}\sum_{j=1}^{m_i} t_k S_{ij} X i_{jk} = T_k$$

$$\sum_{i=1}^{M}\sum_{j=1}^{m_i}\sum_{k=1}^{N} C_k t_k S_{ij} X_{ijk} = b_i$$

$$\sum_{k=1}^{N} X_{ijk} = 1$$

$X_{ijk} = \{0,1\}$ i = 1 , 2 , . . . M

$S_{ij} =\geq 0$ j = 1 , 2 , . . . m_i

$t_k \geq 0$

$C_k > 0$

7. Experimental Result

Let us assume that the Grid system consists of five processors (resources) namely P_1, P_2, P_3, P_4, and P_5 with four sources S_1, S_2, S_3 and S_4 trying to utilize the grid system to execute their workloads. Also assuming that the workloads of the four sources are equally divided and the divisions for S_1 and S_2 become three and for S_3 and S_4 are two respectively. The total workload and the divisions of workloads of each source are shown in table I. Also the processing time, processing cost per unit time and the total available time of each processor are described in table II.

When solving the model using the LINDO software package, the processor assignments are shown in table III. The total cost of the execution of all the workload is Rs.138.

Table 1: Total Workload and its Divisions

Sources	Total Workload	No. of Sub-division
S_1	6	3
S_2	9	3
S_3	6	2
S_4	8	2

Table 2: Processor capacity

Processors	Processing Time	Processing cost	Availability (time unit)
P_1	2	4	12
P_2	3	3	12
P_3	1	2	12
P_4	2	4	12
P_5	4	2	12

Table 3: Processor Allotments

Sources		Processor Allotted
$\mathbf{S_1}$	S_{11}	P_2
	S_{12}	P_3
	S_{13}	P_4
$\mathbf{S_2}$	S_{21}	P_1
	S_{22}	P_1
	S_{23}	P_3
$\mathbf{S_3}$	S_{31}	P_3
	S_{32}	P_5
$\mathbf{S_4}$	S_{41}	P_3
	S_{42}	P_4

8. Conclusion

In this paper, we have proposed scheduling strategies for processing multiple divisible loads on grid systems. As in real life the loads will come dynamically and the resources are also in dynamic in nature. But here we considered the both loads and the resources are in static manner. This work can be extended for dynamic nature also. We are in the process of dynamic arrival of loads and the resources. Also here we have used random number to divide the loads from each source equally. This can be extends to divide the load from a source into the resource capacity. Also we are assumed that there is sufficient buffer space is available in all the resources. The experimental result demonstrates the usefulness of this strategy. We need a system to find out the execution time of a task and also the cost of usage of a processor/resource. The execution time is entirely depends upon the processor speed. Also in this model the number of portion of each total work load of a resource is selected with respect to the random number. So we require a separate module for random number generation. In future it can be modified without using random number as well as unequal division of total workload of each source.

References

[1] V.Bharadwaj, D.Ghose, V.Mani, and T.G. Robertazzi, Scheduling Divisible Loads in Parallel and Distributed Systems. Computer Society Press, Sept.1996.

[2] Han Min Wong, Dantong Yu, Bharadwaj Veeravalli, Thomas G. Robertazzi, “Data Intensive Grid Scheduling: Multiple Sources with Capacity Constraints”, in IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003), Nov.2003.

[3] Kwangil Ko and Thomas G. Robertazzi, “Scheduling in an Environment of Multiple Job Submission”, in Proceedings of the 2002 Conference on Information Sciences and Systems, Mar.2002.

[4] L. Marchal, Y. Yang, H. Casanova, and Y. Robert, “A Realistic Network Application Model for Scheduling Divisible Loads on Large-Scale Platforms,” in International Parallel and Distributed Processing Symposium IPDPS’2005, IEEE Computer Society, Apr. 2005.

[5] M. Moges and T. Robertazzi, “Grid Scheduling Divisible Loads from Multiple Sources via Linear Programming”, in IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2005), 2004.

[6] M. Moges and T. Robertazzi, and D. Yu, “Divisible Load Scheduling with Multiple Sources: Closed Form Solutions”, in Proceedings of 2005 Conference on Information Sciences ans Systems, 2005.

[7] G.Murugesan, C.Chellappan, An Optimal Allocation of Loads from Multiple Sources for Grid Scheduling, in ICETIC’2009, International Conference on Emerging Trends in Computing, Jan.2009

[8] Olivier Beaumont, Arnaud Legrand, and Yves Robert, Scheduling divisible workloads on heterogeneous platforms, Parallel Computing, Vol.29, June2003, pp.1121-1152.

[9] Sivakumar Viswanathan, Bharadwaj Veeravalli, and Thomas G. Robertzzi, Resource-Aware Distributed scheduling Strategies for Large-Scale Computational Cluster/Grid Systems, IEEE Transactions on Parallei and Distributed Systems, Vol.18. No. 10 Oct.2007, pp.1450 -1461.

[10] Thomas G. Robertazzi and Dantong Yu, “Multi-Source Grid Scheduling for Divisible Loads”, in 40th Annual Conference on Information Sciences and Systems, Mar.2006.

G.Murugesan received the B.E. in Computer Science and Engineering from Manonmanium Sundaranar University, Tirunelveli, Tamil Nadu, India in 1996, the M.E. in Systems Engineering and Operations Research from Anna University, Chennai, Tamil Nadu, India in 2006. Presently pursuing Ph.D. in the area of Grid Computing in Anna University, Chennai.

C.Chellappan received his Ph.D in the area of Database Systems from Anna University, Chennai, Tamil Nadu, India where he is currently the Professor and Head of Computer Science and Engineering Department. His research interest includes Mobile Computing, Sensor Networks and Parallel system scheduling. He is also a Principal Investigator of **p**roject(Rs one crore) on “Basic directed Collaborative Research in Smart and Secure Environment ” sponsored by National Technical Research Organization, New Delhi, Govt of India, 2007-2010.

Evidence Gathering System for Input Attacks

Deepak Singh Tomar[1], J.L.Rana [2] and S.C.Shrivastava[3]

[1]Faculty, Department of Computer Science and Engineering,
Maulana Azad National Institute of Technology (MANIT) Bhopal, India
deepaktomar@manit.ac.in

[2]Faculty, Department of Computer Science and Engineering,
Maulana Azad National Institute of Technology (MANIT) Bhopal, India
jl_rana@yahoo.co.in

[3]Faculty, Department of Electronics,
Maulana Azad National Institute of Technology (MANIT) Bhopal, India
scs_manit@yahoo.co.in

Abstract: *In cyber forensic web server logs are an important source for evidence gathering. The user navigation activities on web site are recorded in the web server log file. The attacker exploits web form as an entry point for input attacks like SQL injection, cross site scripting and buffer overflow attack on web application. The web server log does not keep track of the information filled by the end user/attacker in the web form. In this work a prototype system is developed to demonstrate the input attacks and to log the suspicious code (SQL or Script code) fired by attacker to carry out the input attacks to the web application through HTTP.*

Keywords: cyber forensic, input attack, web server log, evidence gathering.

1. Introduction

The input attack is carried out by the suspicious user via entering vulnerable code into the web form or address bar of web browser.

SQL injection, Cross-site scripting (XSS) and buffer overflow are computer security vulnerabilities found in web applications which allow attacker to inject Script / SQL / Values into available web form. SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. In the SQL injection input attack the attacker is inserted arbitrary data, most often a database query, into an available search form that’s eventually executed by the database[1]. The inserted query by attacker may impair the database by retrieving unauthorized data, altering the sensitive data or erasing the data. Both SQL injection and Cross-site scripting (XSS) are the problems of poor web application programming. This form of SQL injection occurs when user input is not filtered for escape characters and is then passed into an SQL statement [3].

Cross-site scripting (XSS) attacks occur when a web server gathers data from a user through web form. A suspicious user may insert tricky java script / VB script code into available web forms which may read and display the current cookie values or redirect the user to another Web site. [3]

In computer buffer memory has a fixed maximum size and is used to store the input data by end user .Buffer overflow input attack is occur when user input exceeds maximum buffer size and extra input goes into unexpected memory locations. In this input attack an attacker insert larger string which may is not accommodate by memory buffer and overflow is occurred. In this way it is easy to crash the web application by overflowing a buffer. Instead of crashing web server attacker is more interested to transfer the control to a suspicious attacker code which may harm the system. [3]

2. Challenges

Limitation of barrier defense (Firewall):- HTTP is considered as a “friendly” traffic by firewall. Generally firewall solutions are ineffective for web application security. The firewall itself is immune to penetration. URL Interpretation attacks, Input Validation attacks, SQL Query Poisoning and HTTP session hijacking can not be prevented by firewall. Firewall is used for direction control; service control, user control and behavior control filter [4]

Missing evidence data in web server log: - Web server logs are an important source of gathering evidence against attacker but it is difficult to discern what truly happened from web logs alone. Web logs may not show if an attack was successful, what happened after an attack and the extent of the attack. In order to discover and understand an attempted web application attack, cyber forensic expert first need to gather all the clues from the crime scene. Collecting these “digital fingerprints” left by the reckless hacker requires that all of the following data fields are available, for every
HTTP request:
- Date
- Time
- Client IP Address
- HTTP Method
- URI
- HTTP Query
- A Full Set of HTTP headers
- The full request body

Some of this data can be extracted from files such as the web server or application server log files, but unfortunately, the most crucial data is unavailable through these sources. Most web and application servers do not grant access to HTTP information such as the full set of HTTP headers and the request body. Without those fields many log entries look alike, and the person conducting the forensics will not be able to distinguish between valid requests and lethal web application attacks [2].

Following code is the example of ““invisible data in HTTP POST request” problem

```
<FORM NAME = M1 METHOD=POST
ACTION=rdata.asp>
Enter login
<INPUT TYPE=TEXTBOX NAME=”tname” >
Enter Password
<INPUT TYPE=PASSWORD NAME=”tpass” >
<INPUT TYPE="SUBMIT" VALUE=”send data”>
</form>
```

Following is an entry of Microsoft IIS log file format

```
2009-09-30 00:15:08 192.168.1.8 - W3SVC1 DEEPAK
192.168.1.4 80 GET /xss1/postprob.asp - 200 0 426 372 594
HTTP/1.1 deepak
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+
SV1) - -
```

All HTTP headers sent by the client. Always prefixed with HTTP_ and capitalized is an actual request capturing HTTP header shown in Figure 1

3. Environmental setup and Experimental Result.

Experiment environment include a Microsoft Internet Information Server (IIS), attacker’s computer and a forensic computer on Institute intranet. (Figure – 2)

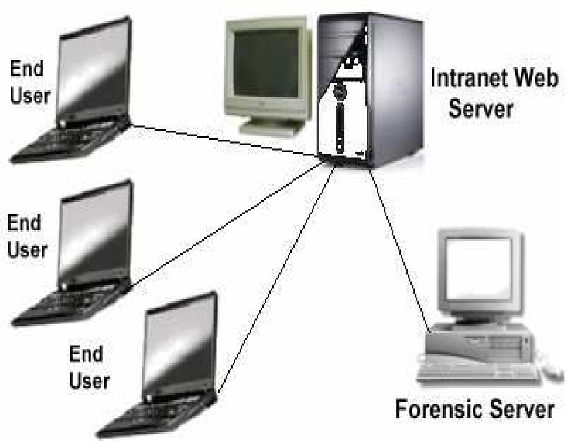
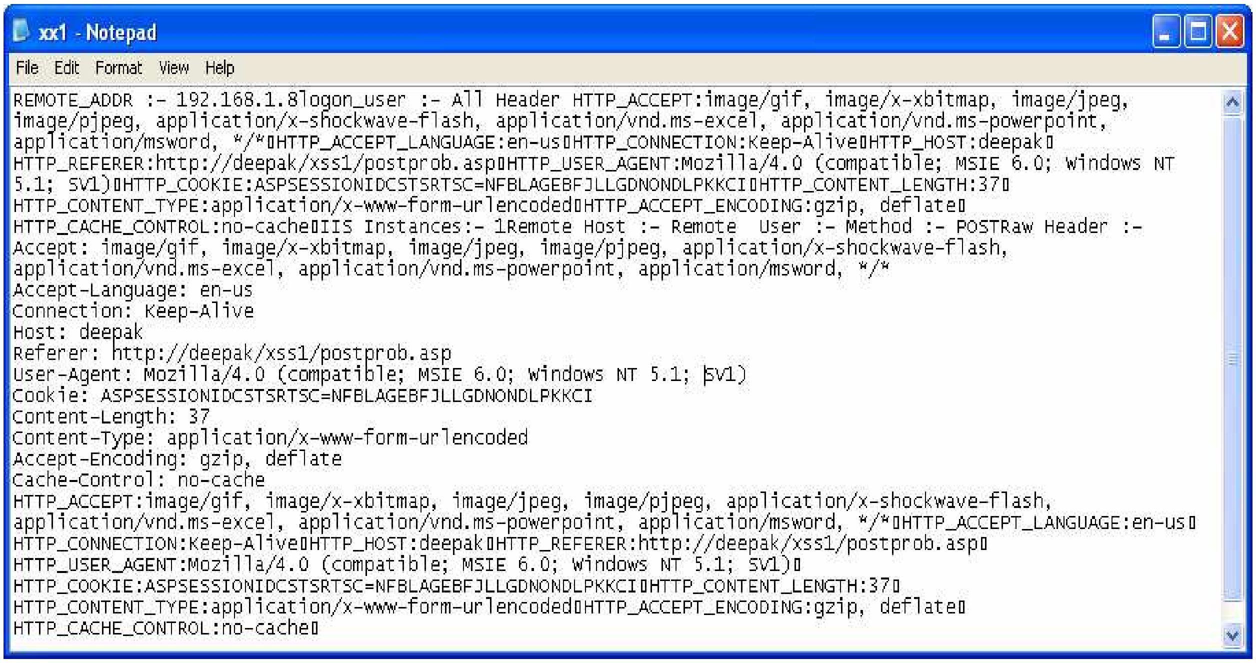


Figure 2: Environmental Setup



```
REMOTE_ADDR :- 192.168.1.8logon_user :- All Header HTTP_ACCEPT:image/gif, image/x-xbitmap, image/jpeg,
image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint,
application/msword, */*▯HTTP_ACCEPT_LANGUAGE:en-us▯HTTP_CONNECTION:Keep-Alive▯HTTP_HOST:deepak▯
HTTP_REFERER:http://deepak/xss1/postprob.asp▯HTTP_USER_AGENT:Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1; SV1)▯HTTP_COOKIE:ASPSESSIONIDCSTSRTSC=NFBLAGEBFJLLGDNONDLPKKCI▯HTTP_CONTENT_LENGTH:37▯
HTTP_CONTENT_TYPE:application/x-www-form-urlencoded▯HTTP_ACCEPT_ENCODING:gzip, deflate▯
HTTP_CACHE_CONTROL:no-cache▯IIS Instances:- 1Remote Host :- Remote  User :- Method :- POSTRaw Header :-
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: en-us
Connection: Keep-Alive
Host: deepak
Referer: http://deepak/xss1/postprob.asp
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Cookie: ASPSESSIONIDCSTSRTSC=NFBLAGEBFJLLGDNONDLPKKCI
Content-Length: 37
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
HTTP_ACCEPT:image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*▯HTTP_ACCEPT_LANGUAGE:en-us▯
HTTP_CONNECTION:Keep-Alive▯HTTP_HOST:deepak▯HTTP_REFERER:http://deepak/xss1/postprob.asp▯
HTTP_USER_AGENT:Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)▯
HTTP_COOKIE:ASPSESSIONIDCSTSRTSC=NFBLAGEBFJLLGDNONDLPKKCI▯HTTP_CONTENT_LENGTH:37▯
HTTP_CONTENT_TYPE:application/x-www-form-urlencoded▯HTTP_ACCEPT_ENCODING:gzip, deflate▯
HTTP_CACHE_CONTROL:no-cache▯
```

Figure 1: Cross side Scripting attack redirecting the control of web site to Hacking Zone

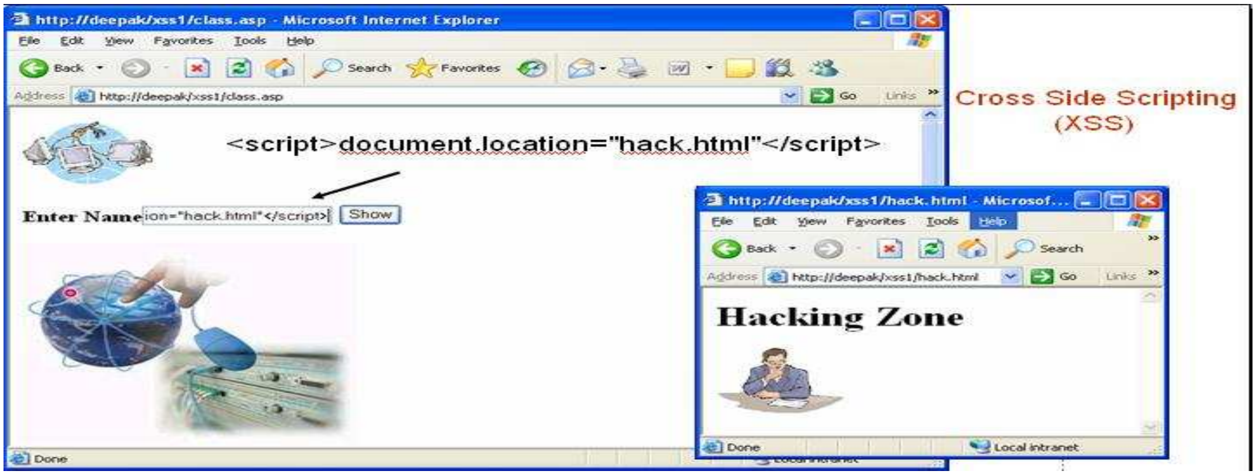
Some of the important data are missing in the log entry. In the web application forensic investigation forensic expert will surely fail to recognize that this request is an input attack by inserting the suspicious data by an attacker into web form.

The environment setup is created using ASP, Jscript and MSACCESS used as database, which is vulnerable to Input attacks. Some attacks such as Cross Side Scripting and SQL injection are performed in this environment to generate attacker scenario.

Cross side scripting attack is shown in the Figure 3; attacker simply inserts following java script code,

```
<script>document.location="hack.html"</script>
```

When submit button is pressed the control is redirected to hack.html


http://deepak/xss1/class.asp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://deepak/xss1/class.asp
Go Links
Cross Side Scripting (XSS)
Show
http://deepak/xss1/hack.html - Microsof...
Address http://deepak/xss1/hack.html
Done
Local intranet

<script>document.location="hack.html"</script>

Enter Name ion="hack.html"</script>

Hacking Zone

Figure 3: Cross side Scripting attack redirecting the control of web site to Hacking Zone

SQL injection attack is shown in the Figure 4 in the normal situation user enters his secret code and relevant information is displayed based on content based filtering. If a suspicious user (attacker) types in secret code field

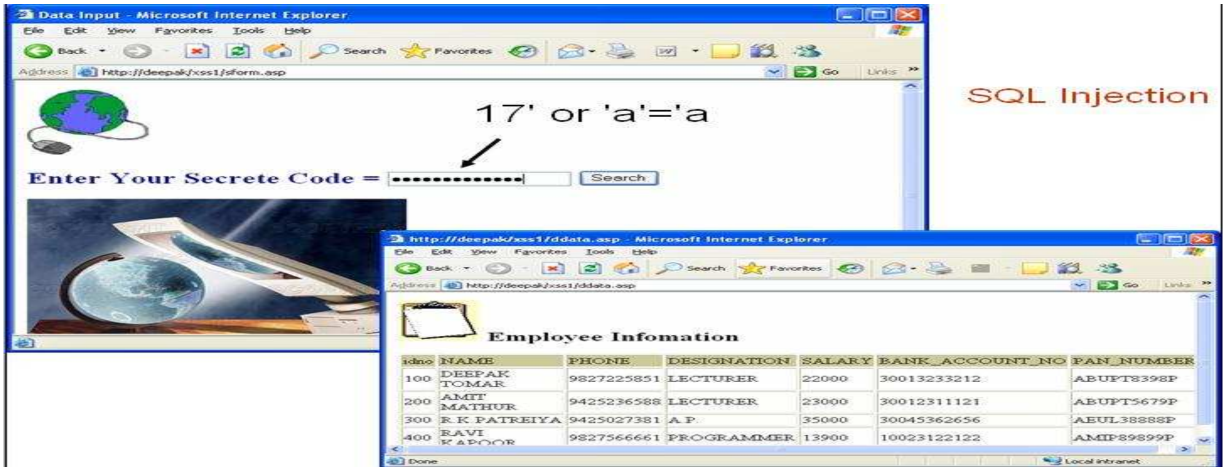
```
17' or 'a'='a
```

This manipulates the server into running the following SQL command:

```
SELECT * from my_employee where scode =
“17” or “a”=”a”;
```

Selects all information stored in the my_employee table. Which is an attack on privacy

As discussed in this paper these attacks using input vulnerability of web application are not recorded in the Microsoft IIS log file. As attacker insert the script / SQL in the web form to conduct input attack it is stored in the developed logging system and is used for gathering evidence. Figure –5 show the log entry of developed system

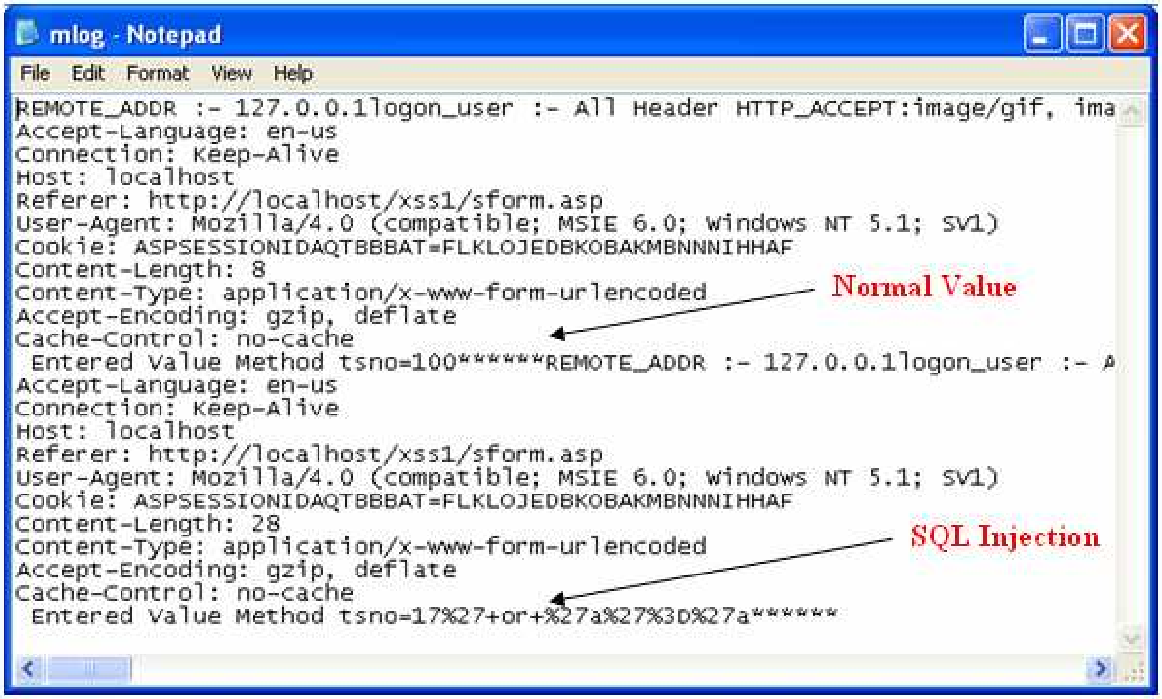

Data Input - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://deepak/xss1/sform.asp
Go Links
SQL Injection
17' or 'a'='a
Search
http://deepak/xss1/ddata.asp - Microsoft Internet Explorer
Address http://deepak/xss1/ddata.asp
Done
Local intranet

Enter Your Secrete Code = ••••••••••••••

Employee Infomation

idno	NAME	PHONE	DESIGNATION	SALARY	BANK_ACCOUNT_NO	PAN_NUMBER
100	DEEPAK TOMAR	9827225851	LECTURER	22000	30013233212	ABUPT8398P
200	AMIT MATHUR	9425236588	LECTURER	23000	30012311121	ABUPT5679P
300	R.K PATREIYA	9425027381	A.P.	35000	30045362656	AEUL38888P
400	RAVI KAPOOR	9827566661	PROGRAMMER	13900	10023122122	AMIP89899P

Figure 4: SQL Injection Attack, providing the attacker with all of the information stored in the my employee table



```
REMOTE_ADDR :- 127.0.0.1logon_user :- All Header HTTP_ACCEPT:image/gif, ima
Accept-Language: en-us
Connection: Keep-Alive
Host: localhost
Referer: http://localhost/xss1/sform.asp
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Cookie: ASPSESSIONIDAQTBBBAT=FLKLOJEDBKOBAKMBNNNIHHAF
Content-Length: 8
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
 Entered Value Method tsno=100******REMOTE_ADDR :- 127.0.0.1logon_user :- A
Accept-Language: en-us
Connection: Keep-Alive
Host: localhost
Referer: http://localhost/xss1/sform.asp
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Cookie: ASPSESSIONIDAQTBBBAT=FLKLOJEDBKOBAKMBNNNIHHAF
Content-Length: 28
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
 Entered Value Method tsno=17%27+or+%27a%27%3D%27a******
```

Figure 5: log generated from developed system showing the normal entry 100 and also show the attacker input string or

4. Conclusion and Future Work

Cyber forensic relies on Web server log events for searching evidence. Web server log files capture the behavior of the web server but not the behavior of the attacker or end user. In this paper a log based evidence gathering system is design and implemented for intranet environment. The developed system also demonstrates the impact points for an input attack scenario that are of prime importance for a forensic investigator. The results are encouraging and authors were able to successfully trace the input attacks from the developed log based system. The developed system primarily gathers evidences for SQL injection and Cross side scripting (XSS) attacks. The system helps the forensic expert to gather the important evidences from the developed log file that was missing in the conventional flat web server log files. The future work shall focus on securing the web server logging system and to improve the structure of web server logs.

References

[1] Karl Scheibelhofer. “SQL Injection Database Security Invalid Input Data”, [Online]. Available: http://www.scribd.com/doc/20856931/L3-SQL-Injection-Invalid-Input-Data [Accessed: Sep 01., 2009].

[2] Web Application Forensics: The Uncharted Territory By Ory Segal, Sanctum Security Group (www.SanctumInc.com) 2002 [Online]. Available www.cgisecurity.com/lib/WhitePaper_Forensics.pdf [Accessed: Sep.10, 2009].

[3] “wilkipedia,” [Online]. Available: http://en.wikipedia.org/wiki [Accessed: Sep.15, 2009]..

[4] “Foundstone,” [Online]. Available: http://www.foundstone.com [Accessed: Sept.17, 2009]..

Deepak Singh Tomar M.Tech & B.E. in Computer Science & Engg. and working as Assistant Professor Computer Science & Engg. Department. Total 14 Years Teaching Experience (PG & UG). Guided 16 M.Tech Thesis.

Dr. J. L . Rana Professor & Head of in Computer Science & Engg. PhD. IIT Mumbai M.S. USA (Huwaii) . Guided Six Ph.D.

Dr. S. C. Shrivastava Professor & Head of Electronics . Guided three Ph.D , 36 M.Tech .Presented nine paper in international & twenty papers in national conference in India

Novel Approach for High Secure Data Hidden in the MPEG Video Using Public Key Infrastructure

A.A Zaidan[1], B.B Zaidan [2]

Department of Computer System & Technology / Faculty of Computer Science and Information Technology/University of Malaya /Kuala Lumpur/Malaysia.50603
aws.alaa@gmail.com, Bilal_Bahaa@hotmail.com

Abstract*:* ***Steganography is the art of information hiding and invisible communication. Unlike cryptography, where the goal is to secure communications from the Snooper by make the data not understood. In this framework we will propose a collaborate approach between steganography and cryptography. This approach will invent high secure data hidden using Public Key Infrastructure (PKI) method. furthermore we will assign the well-built of the PKI algorithm, during this review the author will answer the question why they used PKI algorithm. In additional to the security issues we will use the digital video as a cover to the data hidden. The reason behind opt the video cover in this approach is the huge amount of single frames image per sec which in turn overcome the problem of the data hiding quantity, as the experiment result shows the success of the hidden, encryption, extract, decryption functions without affecting the quality of the video.***

Keyword: Steganography, Hidden Data, Encryption, Decryptio, PKI.

1. Steganography

Steganography is the art of hiding and transmitting data through apparently innocuous carriers in an effort to conceal the existence of the data, the word Steganography literally means covered or hiding writing as derived from Greek. Steganography has its place in security. It is not intended to replace cryptography but supplement it. Hiding a message with Steganography methods reduces the chance of a message being detected. If the message is also encrypted then it provides another layer of protection [1]. Therefore, some Steganographic methods combine traditional Cryptography with Steganography; the sender encrypts the secret message prior to the overall communication process, as it is more difficult for an attacker to detect embedded cipher text in a cover [2]. In the field of Steganography, some terminology has developed. The adjectives 'cover', 'embedded' and 'stego' were defined at the information hiding workshop held in Cambridge, England. The term "cover" refers to description of the original, innocent massage, data, audio, video, and so on. Steganography is not a new science; it dates back to ancient times [3]. It has been used through the ages by ordinary people, spies, rulers, government, and armies [4]. There are many stories about Steganography [5]. For example ancient Greece used methods for hiding messages such as hiding it in the belly of a share (a kind of rabbits), using invisible ink and pigeons. Another ingenious method was to shave the head of a messenger and tattoo a message or image on the messenger head. After allowing his hair to grow, the message would be undetected until the head was shaved again. While the Egyptian used illustrations to conceal message. Hidden information in the cover data is known as the "embedded" data and information hiding is a general term encompassing many sub disciplines, is a term around a wide range of problems beyond that of embedding message in content. The term hiding here can refer to either making the information undetectable or keeping the existence of the information secret. Information hiding is a technique of hiding secret using redundant cover data such as images, audios, movies, documents, etc. This technique has recently become important in a number of application areas. For example, digital video, audio, and images are increasingly embedded with imperceptible marks, which may contain hidden signatures or watermarks that help to prevent unauthorized copy [5]. It is a performance that inserts secret messages into a cover file, so that the existence of the messages is not apparent. Research in information hiding has tremendous increased during the past decade with commercial interests driving the field [5].

2. Cryptography

Cryptography is the art and science of protecting data, which provides ways for converting data into unreadable form using a special key (encrypt it), so that only those who possess a secret key can decipher (or decrypt) the message into plain text. Encrypted messages can sometimes be broken by means of cryptanalysis or code-breaking [6].Cryptography is one of the technological means to provide security to data being transmitted on information and communications systems. In these days since data is usually sent over insure networks such Internet or other public networks Cryptography is required in systems where security of data is concern specially, financial, diplomatic, military or other confidential personal data, whether the data is being transmitted over a medium or kept in a storage device. Cryptography provides a means of verifying the identity of communicating parties, ensuring that only be read only by legitimate parties, ensuring that data reached its destination without modification, preventing the denial of any of communicating parties to have received or sent information [6].Since its first known usage in ancient Egypt Cryptography has passed through different stages and was affected by any major event that affected the way people handled information. In the world war the II for instance Cryptography played an important role and was a key element that gave the allied forces the upper hand, and enables them to win the war sooner, when they were able to dissolve the Enigma cipher machine which the Germans used to encrypt their military secret communications [7].In modern days cryptography is no longer limited to secure sensitive military information but recognized as one of the major components of the security policy of any organization and considered industry standard for providing information security, trust, controlling access to resources, and electronic financial transactions.

2.1 Public Key

Public-key cryptography has been said to be the first truly revolutionary advance in encryption in literally thousands of years [8]. Public Key Cryptography was first described publicly by Stanford University professor Martin Hellman and graduate student Whitfield Diffie in 1976. In their paper "New Directions in Cryptography" they described a two-key crypto system in which two parties could securely communicate over a non-secure communications channel without having to share a secret key. Asymmetric cryptography was born to address the problem of secret key distribution by using two keys instead of a single key. In this process, one key is used for encryption, and the other key is used for decryption. It is called asymmetric because both the keys are required to complete the process. These two keys are collectively known as the key pair. One of the keys (The public key) is freely distributable and used for encryption. Hence, this method of encryption is also called public key encryption. The second key is the secret or private key, is not distributable and is used for decryption. This key, like its name suggests, is private for every communicating entity (Diffie & Hellman, 1976) [7], PKC depends upon the existence of so-called *one-way functions*, or mathematical functions that are easy to computer whereas their inverse function is relatively difficult to compute. Figure 1 depicts the process of encryption and corresponding decryption using public key cryptography.

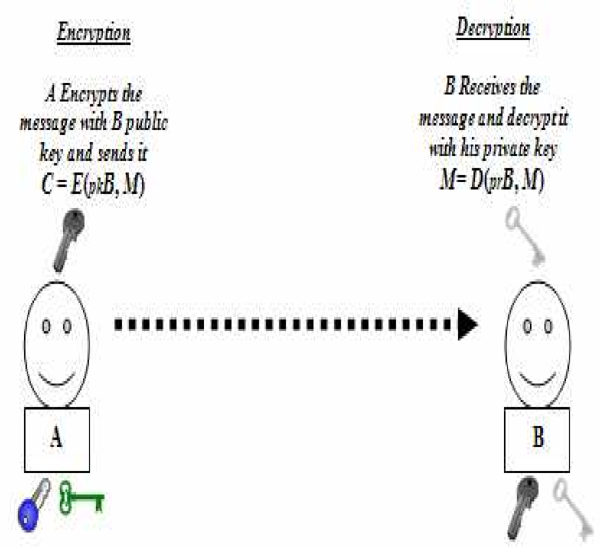

Encryption
A Encrypts the message with B public key and sends it
C = E(pkB, M)
Decryption
B Receives the message and decrypt it with his private key
M= D(prB, M)
A
B

Figure 1: Public Key Encryption

Asymmetric cryptography used two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key. The order in which the two keys are applied is not important, but that both keys are required for the process to work. The sender and receiver do not need to share a key, as required for symmetric encryption. All communications involve only public keys, and no private key is ever transmitted or shared. Every recipient in the above mentioned process will have a unique key that he/she will use to decrypt the data that has been encrypted by its corresponding public key. Public-key cryptography algorithms that are in use today for key exchange or digital signatures and the two most widely used ones are RSA and Diffie Hellman, Digital Signature Algorithm (DSA), Elgamal, and Elliptic Curve Cryptography (ECC).

- *RSA*

According to [7],[6] RSA the most common, PKC implementation, mathematicians who developed it(Ronald Rivest, Adi Shamir, and Leonard Adleman). RSA can be used for key exchange, digital signatures, or encryption of small blocks of data. It uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n, that is the product of two prime numbers chosen according to special rules [6].

Encryption and decryption are of the following form, for some plain text block M and ciphertext block C:
C=Me mod n
M= Cdmod n = (Me) mod n = Medmod n
Both sender and receiver must know the values of n and e, and only the receiver knows the value of d. this make a public key encryption of KU = {e,n} and private of KR {d,n}

- Public Key Infrastructure

Public Key Infrastructure is commonly defined a set of policies, processes, software, hardware, and technologies that use public key cryptography and the certificate management to secure communication. A public key infrastructure (PKI) binds public keys to entities, enables other entities to verify public key bindings, and provides the services needed for ongoing management of keys in distributed systems [9].

PKI integrates digital certificates, public key cryptography, and certification authorities into complete enterprise-wide network security architecture. A typical enterprise's PKI encompasses the issuance of digital certificates to individual users and servers; end-user enrollment software; integration with certificate directories; tools for managing, renewing, and revoking certificates; and related services and support [10].

3. System Overview

The main goal of our plan is to build a system program that is able to hide data in digital video files, more specifically in the images or frames extracted from the digital video file MPEG; as shown in figure 2.

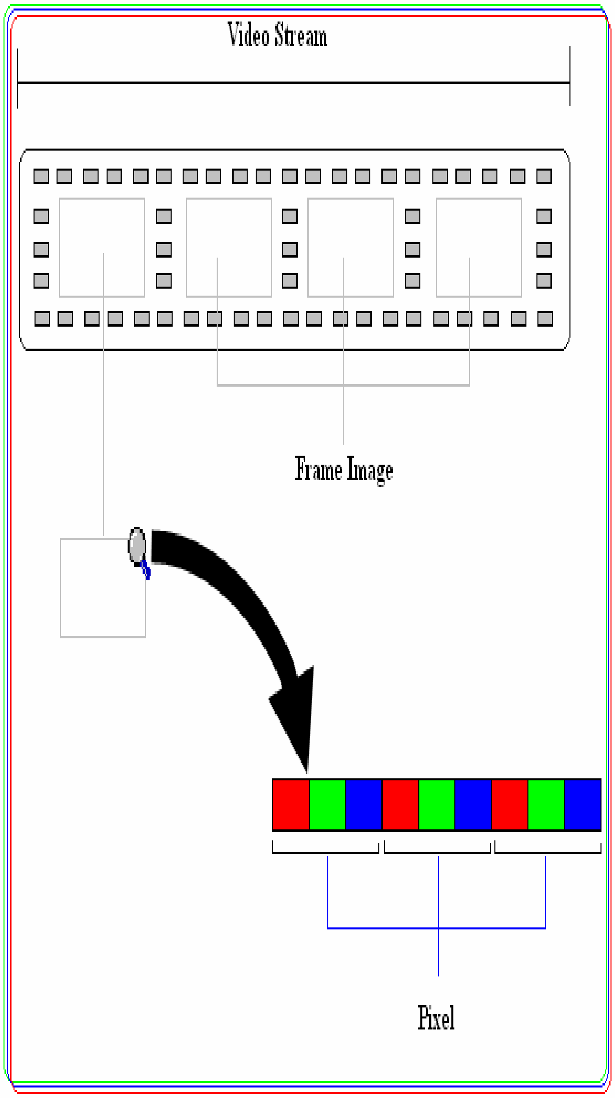


Figure 2: Extracting Frames from Video File

The main function in this framework is steganography and cryptography these two approaches carry out the dreamily protection for the information and make the attackers dream on getting data back. The algorithm work as the chart shows below, where unsuspected carrier with the strongest Asymmetric encryption algorithm building the characteristic of our framework. The main function of the proposed approach is:

- Encrypt data
- Hidden the encrypt data
- Extract the data
- Decrypt the data

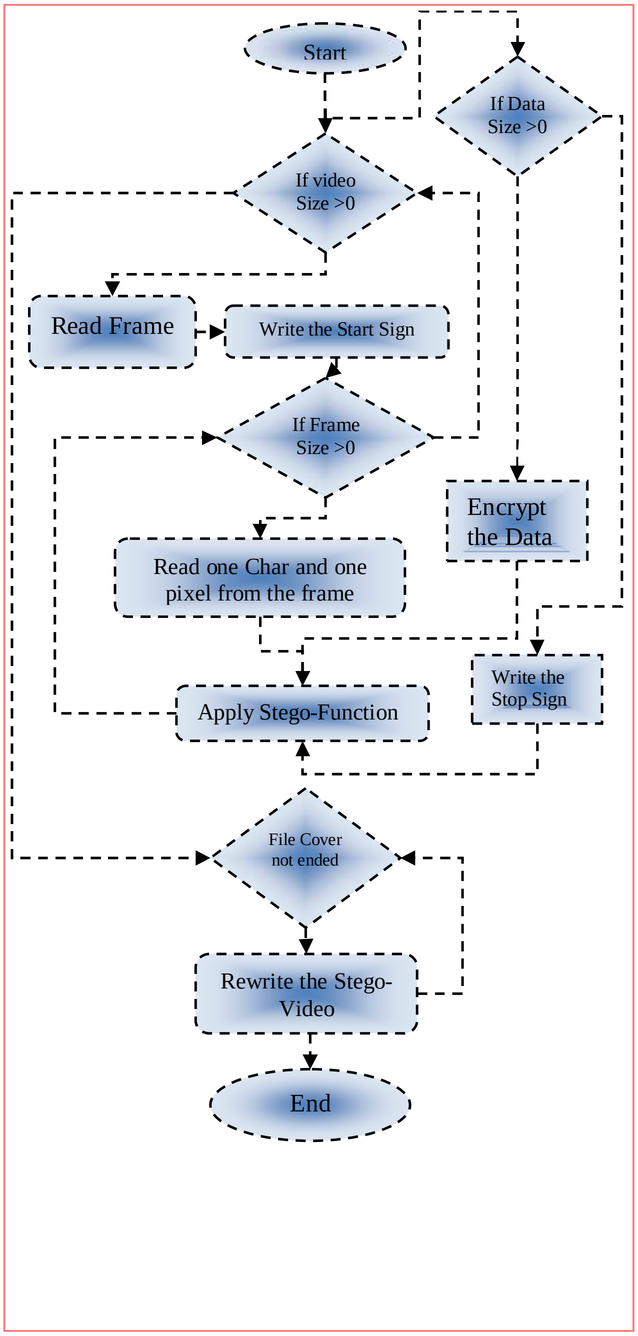


Figure 3: The Encode Algorithm

The figure above showing the encryption with hidden operation this framework give more flexibility to appoint the start point at which frame as well the end point, this new feature make the system more secure in term of avoiding discover the data hidden using the statistical techniques. Figure 4 is the extraction operation with the decryption.

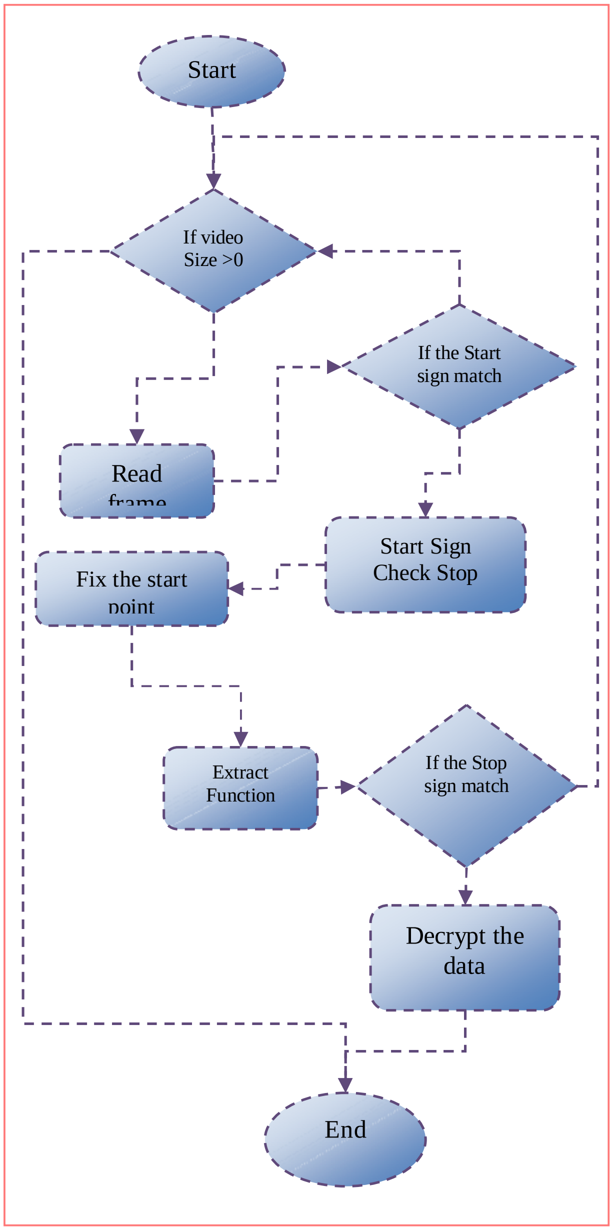


Figure 4: Decoding Algorithm

4. Experimental Results

Due to the difficulty of showing the result as a video stream on paper, the author prefers to display the result on the frame of the digital video file along with histogram of each a single frame. The following here are extracted frames of a digital video file. Figure 5 shows the frames from the famous movie "The Godfather" before applying the algorithm, while Figure 8 shows the frame after applying the algorithm. We can see here that there are no much differences between the two sets of frames especially for human vision system. This can tell that the algorithm can be applied successfully on video frames also to verify the algorithm by the histogram, to see the divergences on the frames before and after hiding data. From the histogram for both single frame in figure 5 & 6, its clear there is no differences between the two sets before and after hiding data which prove that the algorithm successfully hid the data into the frames without making a noticeable difference for the human vision system.

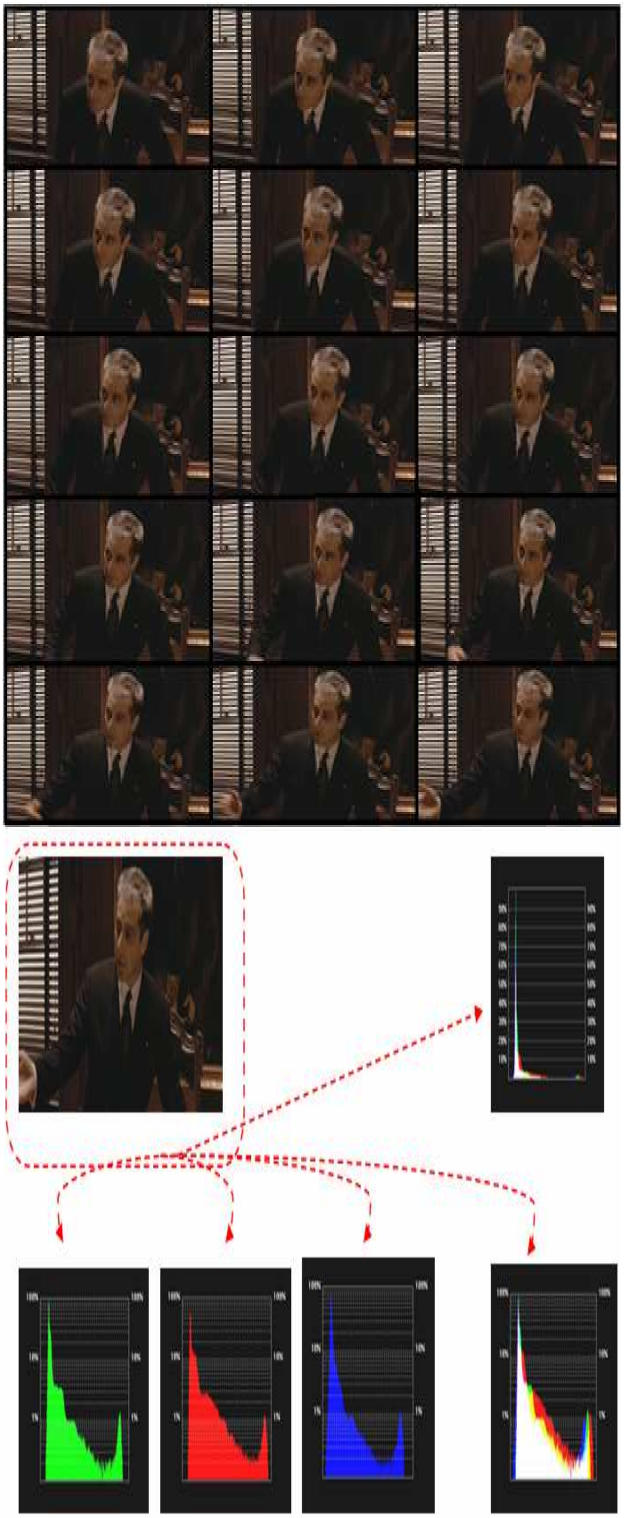


Figure 5: fifteen image frames has taken from a much known movie. "Godfather" before any hidden operation, the first frame under the histogram also the three channels on RGB has been separated for more accuracy on the test

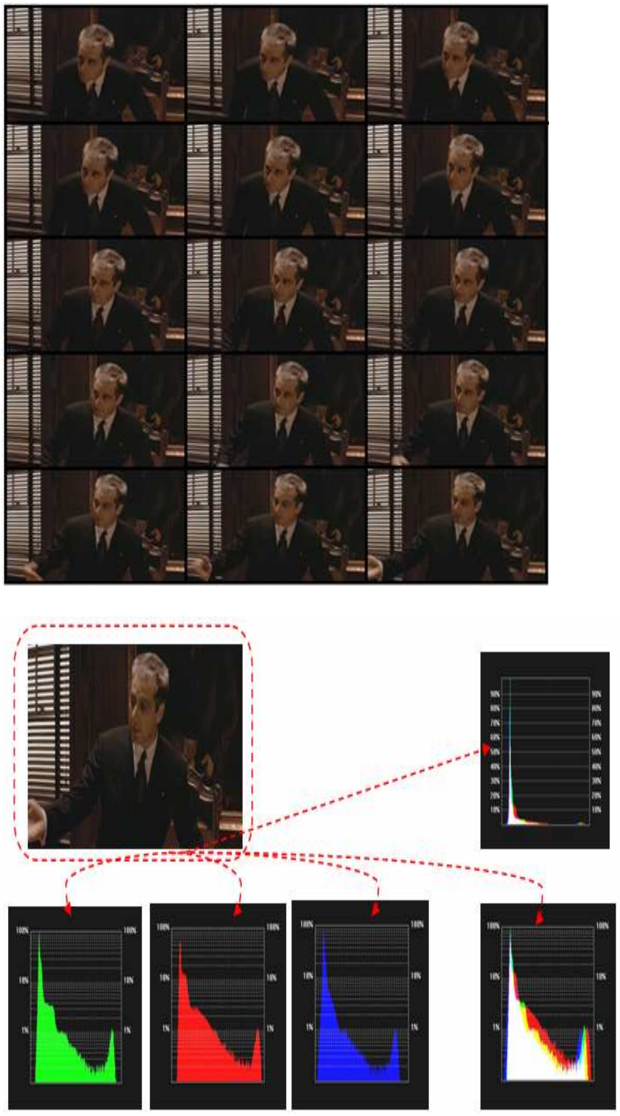


Figure 6: fifteen image frames has taken from a very known movie “Godfather” after hidden operation, the first frame under the histogram also the three channel on RGB has been separated

5. Conclusion

In this paper, a new Approach of high secure video steganography has been invented. The basis of this method is use the digital video as separate frames and hides the information inside. As the experiment result shows the success of the hidden, encryption, extract, decryption function without affecting the quality of the video. This framework overcome the defeat of the limitation of steganography approach by invited the biggest size cover file among the multimedia file which is the video. In the video steganography we have a flexibility of make a selective frame steganography to higher the security of the system or using the whole video too high a huge amount of data hidden. Due the security issues the author has chosen PKI Algorithm method to guarantee the protection of data even the attacker somehow could hold the data.

Acknowledgement

This work was supported in part by the University of Malaya, Kuala Lumpur Malaysia.

References

[1] A.A.Zaidan, B.B.Zaidan, Fazidah Othman, “New Technique of Hidden Data in PE-File with in Unused Area One”, International Journal of Computer and Electrical Engineering (IJCEE), Vol.1, No.5, ISSN: 1793-8198, pp 669-678.

[2] A.A.Zaidan, B.B.Zaidan, Anas Majeed, "High Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm", World Academy of Science Engineering and Technology (WASET), Vol.54, ISSN: 2070-3724, P.P 468-479

[3] Martin Feldhofer, Sandra Dominikus, Johannes Wolkerstorfe “Strong Authentication for RFID Systems Using the AES Algorithm”, springerlink, 2004, ISSN 0302-9743, pp 85-140..

[4] Mohamed Elsadig Eltahir, Laiha Mat Kiah, B.B.Zaidan and A.A.Zaidan," High Rate Video Streaming Steganography", International Conference on Information Management and Engineering (ICIME09), Session 10, P.P 550-553

[5] Fazida.Othman, Miss.Laiha.Maktom, A.Y.Taqa, B.B.Zaidan, A.A.Zaidan, "An Extensive Empirical Study for the Impact of Increasing Data Hidden on the Images Texture", International Conference on Future Computer and Communication (ICFCC 09), Session 7, P.P 477-481

[6] Kessler , Gary C., (1998). An Overview of Cryptography, available,from:http://www.garykessler.net/library/crypto.html#intro. (Accessed April 28, 2008).

[7] Kahn, David , (1980). Cryptology Goes Public, Communications Magazine,IEEE,availablefrom:http://ieeexplore.ieee.org/iel5/35/23736/01090200.pdf?tp= &isnumber=&arnumber=1090200. (Accessed April 28, 2008).

[8] Stallings, William, (2007). Network Security Essentials, applications and Standards p.75, Pearson Education, Inch

[9] Yeob , Chan & Farnham , Tim , (2003). Secure M-Commerce with WPKI, Toshiba Research Europe Limited, Toshiba Telecommunication Research Laboratory, available at: http://www.iris.re.kr/iwap01/program/download/g07_paper.pdf, (accessed May 4, 2008).

[10] Kuhn , D. Richard & Hu ,Vincent C. & Polk , W. Timothy & Chang , Shu-Jen , (2001). Introduction to Public Key Technology and the Federal PKI Infrastructure, National institute of standars and technology,(NIST),available,from:http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf. (Accessed April 28, 2008).

Aos Alaa Zaidan obtained his 1st Class Bachelor degree in Computer Engineering from university of Technology / Baghdad followed by master in data communication and computer network from University of Malaya. He led or member for many funded research projects and He has published more than 40 papers at various international and national conferences and journals, he has done many projects on Steganography for data hidden through different multimedia carriers image, video, audio, text, and non multimedia carrier unused area within exe.file, Quantum Cryptography and Stego-Analysis systems, currently he is working on the multi module for Steganography. He is PhD candidate on the Department of Computer System & Technology / Faculty of Computer Science and Information Technology/University of Malaya /Kuala Lumpur/Malaysia.

Bilal Bahaa Zaidan obtained his bachelor degree in Mathematics and Computer Application from Saddam University/Baghdad followed by master from Department of Computer System & Technology Department Faculty of Computer Science and Information Technology/University of Malaya /Kuala Lumpur/Malaysia, He led or member for many funded research projects and He has published more than 40 papers at various international and national conferences and journals. His research interest on Steganography & Cryptography with his group he has published many papers on data hidden through different multimedia carriers such as image, video, audio, text, and non multimedia careers such as unused area within exe.file, he has done projects on Stego-Analysis systems, currently he is working on Quantum Key Distribution QKD and multi module for Steganography, he is PhD candidate on the Department of Computer System & Technology / Faculty of Computer Science and Information Technology/University of Malaya /Kuala Lumpur/Malaysia.